

Grado en Ingeniería Electrónica Industrial y Automática

2017-2018

Trabajo Fin de Grado

“Sistema electrónico para interferometría con detección síncrona”

Alejandro Javier Ramos San Juan

Tutor

José Antonio García Souto

Leganés, Julio de 2018



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

RESUMEN

En este trabajo se ha diseñado un sistema completo de interferometría que permite realizar mediciones precisas de magnitudes físicas aprovechando las ventajas de uso de la fibra óptica como sensor intrínseco, entre las que destaca su alta sensibilidad.

Para ello se desarrolla una técnica de detección síncrona de la fase óptica en la que a partir de una señal de modulación y de la detección de dos señales en cuadratura (desfasadas entre sí $\pi/2$ rad) se puede obtener la medida deseada, proporcionando una serie de ventajas frente a otras técnicas utilizadas.

El sistema completo está enfocado a la medición precisa de las vibraciones de maquinaria para su monitorización y mantenimiento, pero es posible medir otras magnitudes de frecuencias moderadas (audio y sónar). También puede utilizarse en otros escenarios con esquemas de interferencia óptica, como en la detección sin contacto o para la medición de otros parámetros con la misma configuración del sensor de fibra óptica (temperatura, presión, etc.).

Palabras clave: *interferometría; detección síncrona; fibra óptica; vibraciones; sensor; alta sensibilidad; monitorización; demodulación; fase óptica*

ABSTRACT

In this work a complete interferometry system is designed that allows precise measurements of physical magnitudes taking advantage of the use of optical fiber as intrinsic sensor, which provides a series of advantages among which its high sensitivity stands out.

For this purpose, a technique of synchronous detection of the optical phase is developed in which, from a modulation signal and the detection of two quadrature signals (out of phase $\pi/2$ rad), the desired measurement can be obtained, providing a series of advantages over other techniques used.

The complete system is focused on the precise measurement of the vibrations of machinery for its monitoring and maintenance, but it is possible to measure other magnitudes of moderate frequencies (audio and sonar). It can also be used in other scenarios with optical interference schemes, such as in non-contact detection or for the measurement of other parameters with the same configuration of the optical fiber sensor (temperature, pressure, etc.).

Keywords: *interferometry; synchronous detection; optical fiber; vibrations; sensor; high sensitivity; monitoring; demodulation; optical phase*

AGRADECIMIENTOS

Quiero expresar mis agradecimientos a José Antonio por su confianza y su tiempo dedicado para ayudarme en la elaboración de este trabajo.

Agradezco también el apoyo incondicional de mis padres que siempre han estado a mi lado tanto en los buenos como malos momentos a lo largo de la carrera.

Por último, no me olvido de agradecer todas las personas, compañeros y amigos que me han animado durante este largo camino y la paciencia que han tenido conmigo. Gracias.

TABLA DE CONTENIDO

Resumen.....	iii
Abstract	iv
Agradecimientos	vi
Índice de figuras	ix
Índice de tablas	xi
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura del documento.....	3
2. Estado del arte	4
2.1. Interferencia de la luz.....	4
2.2. Interferómetros.....	7
2.3. Sensores de fibra óptica basados en interferómetros	10
2.4. Técnicas de detección de la fase óptica	13
3. Desarrollo de la técnica de detección síncrona de la fase óptica	15
3.1. Sistema óptico y obtención de la fase óptica.....	15
3.2. Simulación y procesamiento de los resultados	20
3.3. Consideraciones	25
3.4. Sistema completo propuesto	29
4. Diseño e implementación del sistema	31
4.1. Acondicionamiento	31
4.1.1. Conversión de la potencia óptica a señal eléctrica	31
4.1.2. Acondicionamiento de la señal de modulación del PZT.....	33
4.1.3. Acondicionamiento de la señal de salida del interferómetro	35
4.1.4. Placa de circuito impreso	37
4.2. Control y adquisición.....	38
4.2.1. Microcontrolador	39
4.2.2. Generación de la señal de modulación	40
4.2.3. Adquisición de la señal de salida.....	42
4.2.4. Calibración automática	43
4.2.5. Programa, comunicación y conexiones.....	48
4.3. Aplicación informática.....	52

4.3.1.	Tecnologías utilizadas y desarrollo de la aplicación	52
4.3.2.	Uso de la aplicación y visualización de los resultados	53
4.3.3.	Registro de las mediciones.....	59
5.	Ejemplo de aplicación	61
6.	Conclusión y líneas futuras.....	68
7.	Marco regulador.....	70
8.	Entorno socioeconómico.....	71
8.1.	Presupuesto	71
8.2.	Impacto socioeconómico	74
9.	Bibliografía y referencias.....	75
ANEXO I: Algoritmo de detección síncrona de la fase óptica.....		77
Algoritmo.m		77
ANEXO II: Código fuente del microcontrolador		80
Main.c.....		80
Interferometro.h		85
Interferometro.c.....		85
ANEXO III: Protocolo de comunicación mediante USART		96
ANEXO IV: Código fuente de la aplicación informática		100
App.component.html		100
App.component.ts.....		105
ANEXO V: Resumen presentado en el SAAEI 2018		115

ÍNDICE DE FIGURAS

Figura 1. Espectro electromagnético [2]	4
Figura 2. Interferencia constructiva (a) y destructiva (b) de dos ondas sinusoidales ($I_1(t)$ y $I_2(t)$) con intensidad relativa (I_{rel})	6
Figura 3. Franjas de interferencia [4]	7
Figura 4. Esquema óptico del interferómetro Mach-Zehnder [5].....	9
Figura 5. Intensidad relativa a la salida del interferómetro en función de la diferencia de fase óptica entre los dos caminos ópticos.....	9
Figura 6. Diferencias en el camino óptico recorrido: intensidad y fase óptica detectadas	11
Figura 7. Vibraciones simples y compuestas [10]	12
Figura 8. Vibración compuesta originada por los desajustes de una máquina rotatoria [11]	13
Figura 9. Esquema óptico del interferómetro Mach-Zehnder basado en fibra óptica	15
Figura 10. Instalación del interferómetro (sistema óptico) en el Laboratorio de Optoelectrónica (UC3M)	16
Figura 11. Función de transferencia del interferómetro original (azul) y la misma desfasada $\pi/2$ radianes (naranja) sin componente continua (respuesta correspondiente a las dos señales en cuadratura).....	18
Figura 12. Señal de modulación relativa (modulación de la fase óptica) con los máximos (rojo) y mínimos (azul) marcados	21
Figura 13. Señal de salida relativa (salida del interferómetro después de los fotodiodos amplificados balanceados) con las dos señales en cuadratura marcadas (rojo y azul)	22
Figura 14. Fase óptica demodulada: instantánea (línea punteada) y acumulada (línea continua), con el desplazamiento equivalente a la derecha	23
Figura 15. Espectro de frecuencias de la fase óptica acumulada, con el desplazamiento equivalente a la derecha	24
Figura 16. Resultados de aplicar la técnica de detección síncrona a una fase óptica de medida correspondiente a una vibración compuesta	25
Figura 17. Fase óptica detectada muestreando con diferentes números de muestras (N) la fase óptica instantánea.....	26
Figura 18. Error en el cálculo de la tangente y la fase	28
Figura 19. Diagrama de bloques del sistema completo de interferometría	29
Figura 20. Diagrama de bloques del sistema óptico (interferómetro)	30
Figura 21. Diagrama de bloques del sistema electrónico (acondicionamiento).....	31
Figura 22. THORLABS PDB410A con dos fotodiodos y un amplificador de transimpedancia de ganancia fija (fotodiodos amplificados balanceados) [17].....	32
Figura 23. Diagrama funcional de THORLABS PDB410A [17]	32
Figura 24. Alimentación de los circuitos (reguladores de tensión).....	33
Figura 25. Circuito electrónico de acondicionamiento de la señal de modulación del PZT	33
Figura 26. Simulación del circuito de acondicionamiento de la señal de modulación del PZT...	34
Figura 27. Fibra óptica enrollada sobre el PZT	35
Figura 28. Circuito electrónico de acondicionamiento de la señal de salida del interferómetro	35
Figura 29. Simulación del circuito de acondicionamiento de la señal de salida del interferómetro	36

Figura 30. Diseño del layout (a) y componentes (b) de la PCB con dimensiones (ancho x alto): 50.8 x 54.6 mm	37
Figura 31. Resultado final de la PCB	38
Figura 32. Diagrama de bloques del sistema de control y adquisición (procesamiento)	38
Figura 33. Placa de desarrollo NUCLEO-L152RE con el microcontrolador STM32L152RE [18] ..	39
Figura 34. Puntos del periodo de la señal de modulación almacenados (a) y amplitudes disponibles a la salida (b)	42
Figura 35. Sincronización entre ambos timers para adquirir en los máximos y mínimos de la señal de modulación	43
Figura 36. Intensidad de salida relativa del interferómetro calibrado en cuadratura para diferentes fases ópticas iniciales.....	44
Figura 37. Intensidad de salida relativa del interferómetro para diferentes amplitudes de la fase óptica de modulación	45
Figura 38. Intervalos de tiempo máximo y mínimo posibles en la calibración en cuadratura ...	48
Figura 39. Diagrama de flujo del programa principal del microcontrolador	49
Figura 40. Diagrama de flujo de la RAI de la interrupción por recepción asíncrona USART.....	50
Figura 41. Operaciones y procesamiento de la señal de salida	51
Figura 42. Conexiones del microcontrolador utilizadas [18]	51
Figura 43. Diagrama de bloques de la aplicación informática (visualización)	52
Figura 44. Comparación entre los distintos tipos de aplicaciones en la actualidad [20]	53
Figura 45. Pestaña “Configuración” de la aplicación informática.....	54
Figura 46. Conexión del microcontrolador	54
Figura 47. Parámetros de muestreo.....	55
Figura 48. Ajuste de ganancia en adquisición	55
Figura 49. Ajuste de amplitud de modulación	56
Figura 50. Pestaña “Medición” de la aplicación informática	56
Figura 51. Gráficos intermedios (desplazamiento de fase constante de 0 a 2π radianes)	57
Figura 52. Gráfico de la FFT.....	58
Figura 53. Configuración de la adquisición programada.....	58
Figura 54. Pestaña “Historial” de la aplicación informática.....	59
Figura 55. Formato de los datos guardados en JSON	60
Figura 56. Comunicación serie entre el microcontrolador (verde) y el ordenador (azul)	61
Figura 57. Señales de modulación V_i (acondicionada de la proporcionada por el microcontrolador y que se aplica al PZT) y de salida del interferómetro V_o (acondicionada para ser adquirida por el microcontrolador).....	62
Figura 58. Transformador eléctrico de baja potencia (a) con la fibra óptica enrollada adherida al núcleo del transformador (b)	63
Figura 59. Gráficos intermedios obtenidos en la medición del transformador	63
Figura 60. Gráfico de la FFT de la medición del transformador.....	64
Figura 61. Comprobación de la calibración manual del interferómetro (las dos señales en los máximos y mínimos se encuentran en cuadratura).....	65
Figura 62. Resultados obtenidos manualmente en la medición de las vibraciones del transformador	66
Figura 63. Resultados obtenidos del acelerómetro en la medición de las vibraciones del transformador	67

ÍNDICE DE TABLAS

Tabla 1. Relación entre los signos de las señales en cuadratura, los rangos de fase y la fase ajustada	22
Tabla 2. Componentes electrónicos de la PCB.....	37
Tabla 3. Atributos de los datos guardados de cada medición	59
Tabla 4. Presupuesto	71
Tabla 5. Resumen del presupuesto	73
Tabla 6. Formato de los datos enviados al microcontrolador.....	96
Tabla 7. Formato de los datos recibidos del microcontrolador	96
Tabla 8. Tipos de respuesta.....	97
Tabla 9. Comandos del programa del microcontrolador	97

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

En la actualidad, cada vez es más necesario una precisión y flexibilidad excelentes en la medida de magnitudes físicas. El control preventivo y mantenimiento de la maquinaria en procesos de producción son dos aspectos fundamentales a tener en cuenta para obtener el máximo rendimiento y calidad en el producto final, siendo necesario monitorizar sus parámetros para evaluar su correcto funcionamiento y detectar posibles fallos.

Típicamente, la medida de parámetros físicos se realiza mediante sensores y transductores (galgas extensiométricas, por ejemplo), existiendo varios tipos de estos en función de la magnitud física a la que son sensibles y se desea medir. Sin embargo, el uso de estos transductores da lugar a una serie de inconvenientes como la necesidad de estar en contacto permanente con la máquina que se desea controlar. Esto implica que el sensor se somete a las mismas vibraciones y a otros factores ambientales como la temperatura, el calor emitido por la máquina o campos magnéticos, siendo por ende más propenso al desgaste, a su ruptura y también a medidas erróneas. Además, su sensibilidad y respuesta en frecuencia depende del tamaño y formato de estos.

Por estas razones, la interferometría basada en sensores de fibra óptica está ganando cada vez más importancia en los últimos años. El uso de la fibra óptica como sensor intrínseco proporciona varias ventajas frente a otros tipos de sensores entre las que destacan su alta sensibilidad y su inmunidad frente a algunos factores externos. Sin embargo, para aprovecharse de sus ventajas es necesario recurrir a técnicas que en ocasiones pueden resultar complejas y costosas de implementar.

En este trabajo se desarrolla un sistema completo de interferometría que permite realizar mediciones precisas de magnitudes físicas aprovechando las propiedades de la fibra óptica y el fenómeno de interferencia de la luz. Para ello se desarrolla una técnica de detección síncrona de la fase óptica en la que a partir de dos señales en cuadratura se puede obtener la medida deseada y además proporciona varias ventajas sobre otras técnicas utilizadas. En principio, el sistema estará enfocado en la medición precisa de vibraciones de maquinaria con contacto, siendo posible medir frecuencias moderadas (audio y sónar), pero puede ser utilizado para otros fines entre los que se incluye la medición sin contacto o la medición de temperatura, por ejemplo. Sus principales ventajas frente a otro tipo de sensores son la alta sensibilidad y resolución en la medida, además de un muy buen margen dinámico.

1.2. OBJETIVOS

El objetivo principal del presente trabajo es el desarrollo de un sistema electrónico completo para el acondicionamiento de sistemas de interferometría láser, desde la modulación de la fase óptica a través de un actuador piezoeléctrico hasta la visualización de los resultados obtenidos en la medición y el registro de los mismos. El sistema podrá ser utilizado posteriormente para la medición precisa de vibraciones o desplazamientos, entre otras magnitudes físicas.

Primero se realizará el montaje del sistema óptico que actuará como sensor de las magnitudes físicas basándose en el principio de interferencia de la luz, obteniendo una salida característica en función de la amplitud y la frecuencia de las magnitudes sensibles.

A continuación, se desarrollará una técnica de detección síncrona que permita obtener el valor de dichas magnitudes y que presentará varias ventajas frente a otras técnicas existentes. Después se realizarán ensayos y simulaciones que permitan conocer el comportamiento de la misma en diferentes escenarios, mostrando finalmente un ejemplo de aplicación.

El proyecto continuará con el desarrollo de un sistema de control y adquisición a través de un microcontrolador que permita manejar el sistema, realizar mediciones y obtener los resultados procesados de forma prácticamente instantánea, diseñando un protocolo de comunicación entre éste y el ordenador.

Junto al sistema de control y adquisición, se diseñará un circuito de acondicionamiento electrónico que acondicione las señales de entrada al sistema óptico y de salida al microcontrolador para adquirir y procesar la respuesta correctamente.

Una vez simulado y probado el funcionamiento del circuito de acondicionamiento, se diseñará una placa de circuito impreso que permita la posterior portabilidad del sistema o su uso en un sistema óptico distinto.

Finalmente, también se diseñará una aplicación informática portable con interfaz gráfica mediante la cual se podrá controlar el sistema y visualizar los resultados de las mediciones realizadas, además de guardar un registro de las mismas.

1.3. ESTRUCTURA DEL DOCUMENTO

La estructura del documento es la siguiente:

- **Capítulo 1 – Introducción:** se explican la motivación y los objetivos del proyecto realizado, así como la organización del presente documento.
- **Capítulo 2 – Estado del arte:** se describen los conceptos de teoría necesarios para abordar el proyecto, y trabajos o estudios existentes relacionados con el mismo.
- **Capítulo 3 – Desarrollo de la técnica de detección síncrona de la fase óptica:** en este capítulo se desarrolla la técnica utilizada para obtener la medida de interés mediante interferometría con fibra óptica.
- **Capítulo 4 – Diseño e implementación del sistema:** se detallan los procesos de análisis, diseño e implementación del sistema completo de interferometría desarrollado que integra la técnica anterior.
- **Capítulo 5 – Ejemplo de aplicación:** en este capítulo se realiza una prueba de funcionamiento general de todo el sistema en un caso real.
- **Capítulo 6 – Conclusión y líneas futuras:** se expone la reflexión final del trabajo realizado, los objetivos cumplidos y posibles futuras mejoras.
- **Capítulo 7 – Marco regulador:** se incluye un breve estudio sobre la legislación aplicable, estándares técnicos, patentes y propiedad intelectual.
- **Capítulo 8 – Entorno socioeconómico:** se detalla el presupuesto del trabajo realizado y un breve análisis del impacto socioeconómico.
- **Capítulo 9 – Bibliografía y referencias:** se recogen los medios consultados y utilizados para realización del trabajo.
- **ANEXOS**

2. ESTADO DEL ARTE

2.1. INTERFERENCIA DE LA LUZ

La luz puede considerarse como una onda electromagnética transversal propagándose a través del espacio. Debido a la relación entre el campo magnético y el campo eléctrico (perpendiculares entre sí) y su propagación conjunta, normalmente es suficiente para caracterizar la luz considerando solo el campo eléctrico en cualquier punto. El campo eléctrico puede representarse como un vector perpendicular a la dirección de propagación de la onda. Si este vector se encuentra en un mismo plano en todo momento, entonces la luz se encuentra linealmente polarizada en ese plano y se puede describir el campo eléctrico asociado a la propagación de luz en cualquier punto mediante la siguiente ecuación (considerando una onda plana propagándose en una dirección a lo largo del eje z) [1]:

$$E(z, t) = a \cdot \cos\left(2\pi\left(ft - \frac{z}{\lambda}\right)\right) = a \cdot \cos(\omega t - kz) \quad (2.1)$$

donde z es la distancia recorrida, a es la amplitud de la onda, f es su frecuencia temporal, λ es su longitud de onda, $\omega = 2\pi f$ es su frecuencia angular y $k = 2\pi/\lambda$ es la constante de propagación.

La luz visible comprende las longitudes de onda en el vacío desde los 400 nm (violeta) a los 750 nm (rojo) del espectro electromagnético (**Figura 1**), con unas frecuencias aproximadas asociadas de $7.5 \cdot 10^{14} \text{ Hz}$ y $4.0 \cdot 10^{14} \text{ Hz}$ respectivamente, que definen el color. Las longitudes de onda menores corresponden a la región ultravioleta (UV) mientras que las longitudes de onda mayores corresponden a la región infrarroja (IR).

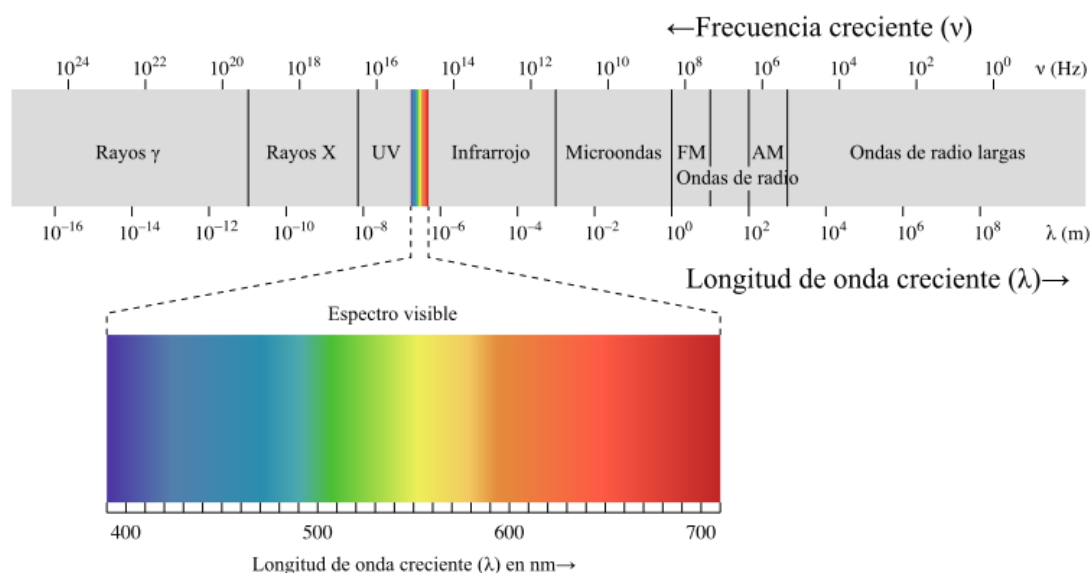


Figura 1. Espectro electromagnético [2]

En un medio distinto del vacío con un índice de refracción n , teniendo en cuenta que la frecuencia f permanece constante, la longitud de onda λ_n en ese medio es:

$$\lambda_n = \frac{\lambda}{n} \quad (2.2)$$

Además, una superficie de fase constante (también llamada frente de onda), se mueve en el eje z a una velocidad en el vacío c :

$$c = \lambda f = 3 \cdot 10^8 \text{ m/s} \quad (2.3)$$

Por otro lado, la interferometría es una técnica que hace uso del principio de superposición de manera que, en la combinación de dos ondas, la onda resultante proporciona la información útil para obtener una medida deseada. La interferometría óptica, concretamente, se basa en la superposición de luz.

Algunas de las aplicaciones actuales de la interferometría óptica son: medición de distancias, desplazamientos y vibraciones; comprobación del funcionamiento de sistemas ópticos; estudios sobre flujos de gas y plasmas; estudios de la superficie topográfica; medición de temperatura, presión y campos eléctricos o magnéticos; detección de rotación; espectrografía de alta resolución, etc. Existen otras aplicaciones en investigación, entre las que destaca la lógica totalmente óptica de alta velocidad (*high-speed all-optical logic*). Destaca también el reciente descubrimiento de la existencia de ondas gravitacionales en tres lugares diferentes mediante técnicas de interferometría [3].

Cuando dos ondas de luz se superponen, la intensidad óptica resultante en un punto depende de la cancelación o el refuerzo entre ellas. La suma de amplitudes es:

$$\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2 = a_1^{-i\phi_1} + a_2^{-i\phi_2} \quad (2.4)$$

donde \mathbf{A}_1 y \mathbf{A}_2 son las amplitudes complejas de las respectivas ondas.

La intensidad óptica final es proporcional al cuadrado de la suma de las amplitudes:

$$\begin{aligned} |\mathbf{A}|^2 &= \mathbf{A}\bar{\mathbf{A}} = (\mathbf{A}_1 + \mathbf{A}_2)(\bar{\mathbf{A}}_1 + \bar{\mathbf{A}}_2) = |\mathbf{A}_1|^2 + |\mathbf{A}_2|^2 + \mathbf{A}_1\bar{\mathbf{A}}_2 + \bar{\mathbf{A}}_1\mathbf{A}_2 \\ I &\propto |\mathbf{A}|^2 \rightarrow I = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\Delta\phi) \end{aligned} \quad (2.5)$$

donde I_1 y I_2 son las intensidades de las respectivas ondas y $\Delta\phi = \phi_1 - \phi_2$ es la diferencia de fase óptica.

De esta ecuación pueden destacarse dos casos especiales: interferencia constructiva e interferencia destructiva. Considerando dos ondas sinusoidales con la misma amplitud, en el primer caso ambas ondas se encuentran en fase $\Delta\phi = 0^\circ$ (los valles y crestas coinciden en ambas ondas), resultando en una onda con el doble de amplitud de cualquiera de ellas. Por el contrario, en el segundo caso las ondas están desfasadas $\Delta\phi = 180^\circ$ o en contrafase (los valles y crestas de ambas ondas no coinciden), resultando en la cancelación entre ellas, o lo que es lo mismo, en una onda de amplitud nula (**Figura 2**).

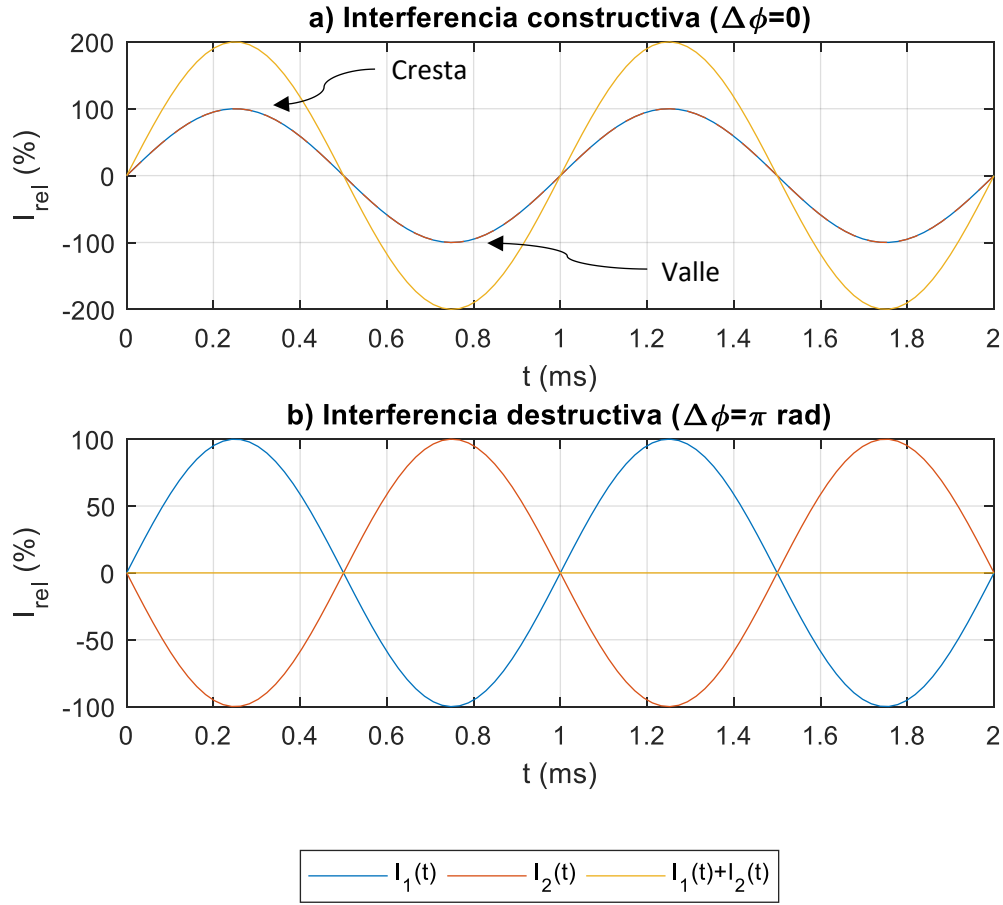


Figura 2. Interferencia constructiva (a) y destructiva (b) de dos ondas sinusoidales ($I_1(t)$ y $I_2(t)$) con intensidad relativa (I_{rel})

Si las dos ondas derivan de una fuente común de manera que presenten la misma fase óptica de origen, entonces la diferencia de fase $\Delta\phi$ corresponde a una diferencia en el camino óptico $\Delta\beta$:

$$\Delta\beta = \left(\frac{\lambda}{2\pi}\right) \Delta\phi \quad (2.6)$$

que también se puede observar como un retraso en el tiempo $\Delta\tau$:

$$\Delta\tau = \frac{\Delta p}{c} = \left(\frac{\lambda}{2\pi c}\right) \Delta\phi \quad (2.7)$$

Estas dos últimas relaciones son el principio de medida de la interferometría. Otro parámetro importante es el orden de interferencia N . Conociendo el orden de interferencia se puede obtener una diferencia de caminos superior a la que se obtendría con una variación en fase de 2π radianes en módulo. Es decir, se puede continuar midiendo teóricamente una distancia infinita acumulando los valores anteriores. Su valor es:

$$N = \frac{\Delta\phi}{2\pi} = \frac{\Delta p}{\lambda} = f\Delta\tau \quad (2.8)$$

Si la diferencia de fase $\Delta\phi$ entre los dos haces varía linealmente a lo largo del campo de visión, la intensidad varía sinusoidalmente originando “bandas claras y oscuras” conocidas como franjas de interferencia, y su conjunto como patrón de interferencia (**Figura 3**). Las bandas claras se corresponden a la interferencia constructiva, mientras que las bandas oscuras se corresponden a la interferencia destructiva.

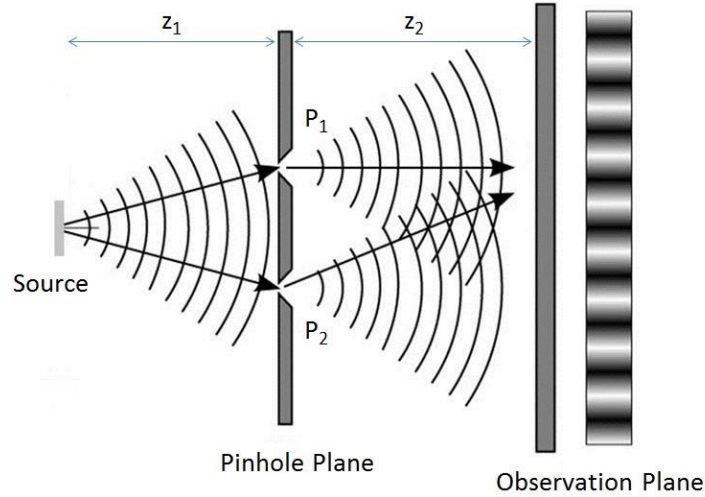


Figura 3. Franjas de interferencia [4]

La visibilidad V de estas franjas de interferencia se define como:

$$V = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \quad (2.9)$$

siendo $0 \leq V \leq 1$. Las intensidades máxima y mínima (I_{max} y I_{min}) se calculan sustituyendo $\Delta\phi = 2m\pi$ y $\Delta\phi = (2m + 1)\pi$ respectivamente (con $m \in \mathbb{N}$) en la ecuación (2.5):

$$\left. \begin{aligned} I_{max} &= I_1 + I_2 + 2 \cdot \sqrt{I_1 I_2} \\ I_{min} &= I_1 + I_2 - 2 \cdot \sqrt{I_1 I_2} \end{aligned} \right\} \rightarrow V = \frac{2\sqrt{I_1 I_2}}{I_1 + I_2} \quad (2.10)$$

Combinando las ecuaciones (2.5) y (2.10) se obtiene:

$$I = I_1 + I_2 + V(I_1 + I_2) \cos(\Delta\phi) \xrightarrow{I_0 = I_1 + I_2} I = I_0(1 + V \cos(\Delta\phi)) \quad (2.11)$$

donde I_0 es la intensidad óptica proporcionada por la fuente.

De esta manera, se obtiene la intensidad óptica en función de la diferencia de fase óptica en cualquier punto.

2.2. INTERFERÓMETROS

Para realizar mediciones precisas mediante interferencia de la luz se precisa de un montaje en el que dos haces que viajan a través de dos caminos diferentes se hacen interferir: el interferómetro. Uno de estos caminos ópticos es el camino de referencia (β_{ref}) mientras que el otro es el camino de medida (β_{med}). La diferencia en el camino óptico $\Delta\beta$ es entonces:

$$\Delta\beta = \beta_{med} - \beta_{ref} \quad (2.12)$$

Para obtener un patrón de interferencia estacionario, la diferencia de fase óptica entre los dos haces que viajan por ambos caminos no debe cambiar con el tiempo. Es decir, ambos haces deben tener la misma frecuencia. Como ya se ha mencionado, esto es equivalente a que ambos haces deben proceder de la misma fuente. Los dos métodos para obtener dos haces a partir de una sola fuente son: división del frente de onda y división de la amplitud. En la división del frente de onda se usan dos aperturas para aislar dos haces a partir de dos porciones separadas del frente de onda primario. En la división de la amplitud, técnica utilizada en el presente trabajo, los dos haces derivan de la misma porción del frente de onda original. Para ello se pueden utilizar varios elementos ópticos. Concretamente, en este trabajo se utilizará un separador de haz (*beam splitter*).

Existen numerosos tipos de interferómetros a partir de los cuales se obtiene un patrón de interferencia distinto. Los interferómetros de dos haces más comunes son:

- **Interferómetro Rayleigh:** hace uso de la división del frente de onda. Su aplicación más común es la medida del índice de refracción de un gas. Tiene como ventajas principales la simplicidad, la estabilidad y la posibilidad de utilizar fuentes de luz no coherentes (como la luz blanca). Sin embargo, para observar bien las franjas de interferencia deben ampliarse considerablemente y se debe utilizar una fuente de luz puntual o lineal.
- **Interferómetro Michelson:** el haz primario de la fuente se divide mediante un separador de haz que también es utilizado para posteriormente recombinar los rayos reflejados en los dos espejos. Es un interferómetro fácil de instalar y de alinear. Sus aplicaciones más comunes son la medida de longitud y la comprobación del funcionamiento de sistemas ópticos.
- **Interferómetro Mach-Zehnder:** hace uso de dos separadores de haz y de dos espejos para dividir y recombinar los haces de luz. Destaca por presentar dos caminos que se encuentran separados y atravesados una sola vez por los haces de luz, además de poder hacer coincidir la región de las franjas de interferencia con el objeto de prueba. Es ampliamente utilizado en estudios de flujo de fluidos, transferencia de calor y distribución de temperatura en plasmas.
- **Interferómetro Sagnac:** se trata de un interferómetro de camino común (a diferencia de los tres anteriores) en el que los dos haces atraviesan el mismo camino en direcciones opuestas, obteniendo dos importantes ventajas. Las vibraciones y corrientes de aire no afectan en gran medida a la estabilidad de las franjas de interferencia. Además, es un interferómetro muy fácil de alinear y muy estable. Presenta aplicaciones en la detección de rotación sustituyendo a los giroscopios convencionales.

Para el propósito de este trabajo se ha escogido el interferómetro Mach-Zehnder, cuyo esquema óptico se muestra en la **Figura 4**.

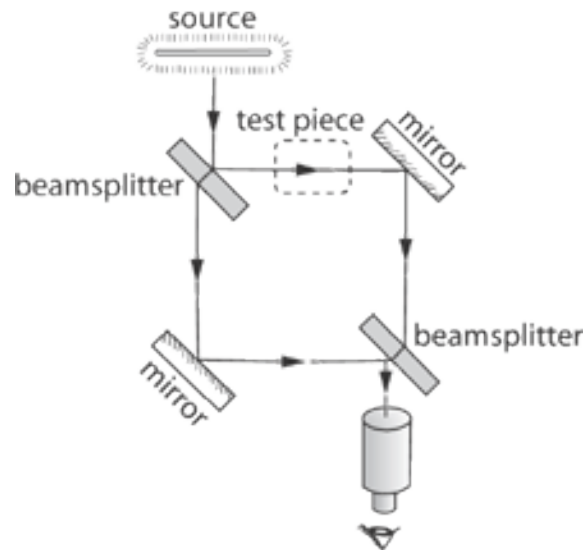


Figura 4. Esquema óptico del interferómetro Mach-Zehnder [5]

Considerando una visibilidad máxima ($V = 1$), se puede representar la respuesta a la salida del interferómetro (intensidad I_{rel} relativa a la intensidad proporcionada por la fuente) en función de la diferencia de fase $\Delta\phi$ suponiendo que no existen pérdidas de intensidad óptica y que el interferómetro se encuentra totalmente equilibrado (**Figura 5**). A partir de la función de transferencia del interferómetro representada, correspondiente a la ecuación (2.11), y de la relación (2.6) se puede obtener la medida física de interés.

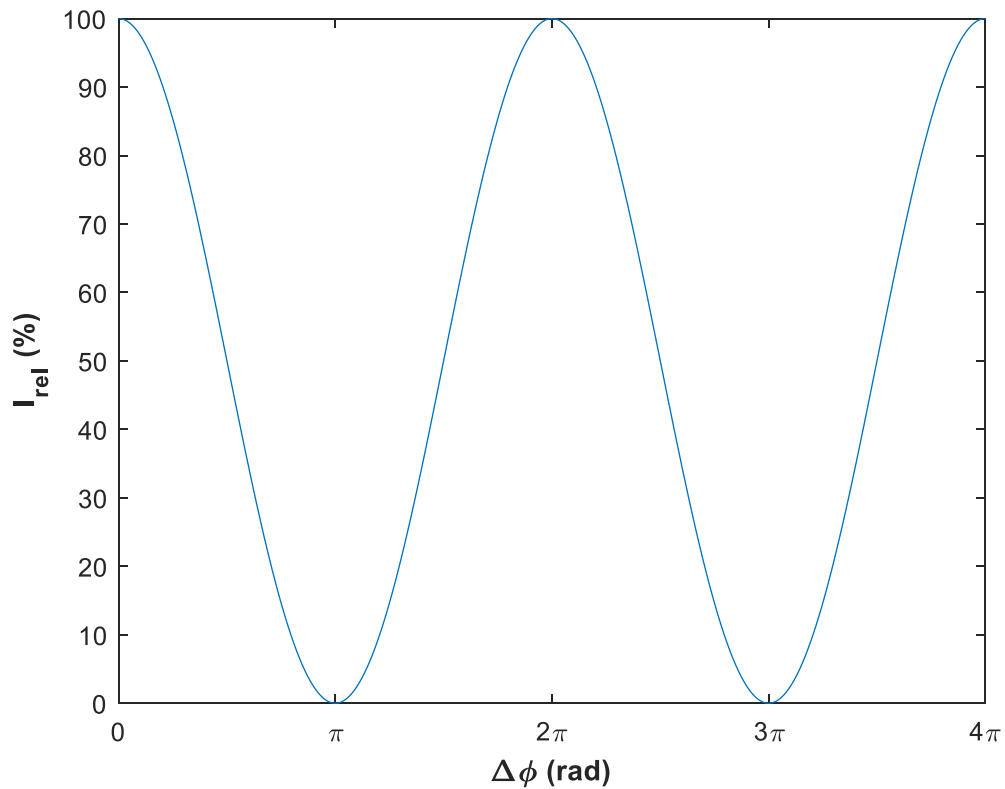


Figura 5. Intensidad relativa a la salida del interferómetro en función de la diferencia de fase óptica entre los dos caminos ópticos

2.3. SENSORES DE FIBRA ÓPTICA BASADOS EN INTERFERÓMETROS

El interferómetro puede ser implementado en espacio libre o mediante fibra óptica. La fibra óptica ha llamado la atención en varios campos: comunicaciones, biología, química (caracterización de líquidos), mecánica (medida de tensión), etc. La fibra óptica es un medio de transmisión de luz basado en los principios de reflexión, de manera que el haz de luz queda totalmente confinado dentro de ella propagándose con un ángulo de reflexión por encima del ángulo límite o crítico de reflexión total (reflexión interna total). Sin embargo, el interés de uso de la fibra óptica en este caso no se debe a las propiedades excelentes de transmisión de la luz, sino a la capacidad de cambiar sus propiedades en función de las condiciones externas comportándose como un transductor intrínseco.

Comparado con métodos eléctricos y mecánicos tradicionales, los sensores de fibra óptica presentan numerosas ventajas como la inmunidad electromagnética, la resistencia a la erosión, la alta sensibilidad, la capacidad de detección o la medida remota sin precisar una visión directa con el objeto de medida. Por ello, en los últimos años se han desarrollado diversos tipos de sensores de fibra óptica [6]. En concreto, los sensores de fibra óptica basados en interferómetros Mach-Zehnder han recibido una considerable atención en campos como la medida de temperatura [7] o en vibrometría con y sin contacto [8], siendo compactos, robustos y con una gran flexibilidad en su configuración.

Aunque existen varias formas de clasificar los sensores de fibra óptica, se pueden distinguir dos grupos en función del recorrido de la luz y de dónde se produce la transducción [9]:

- **Sensores extrínsecos:** la luz abandona la fibra óptica y se refleja en el objeto de medida antes de volver al sistema de fibra óptica. Presenta como inconvenientes que la medida puede verse afectada por polvo, vibraciones o alineamiento.
- **Sensores intrínsecos:** la luz permanece en la fibra óptica durante toda la medición, evitando así los anteriores problemas mencionados. De esta manera, se miden también los efectos ambientales en la señal óptica a lo largo de la fibra óptica a través de la que viaja. Estos sensores pueden medir temperatura, presión, deformación o tensión (del material) entre otros parámetros ajustando la fibra de manera que la cantidad medible modula la intensidad, fase, polarización, longitud de onda o tiempo de recorrido de la luz en la fibra óptica. Además, la flexibilidad de la fibra óptica permite configurar el sensor a medida: puntual o distribuido (continuo, discreto o media), mediante redes de Bragg en fibra (FBG, *Fiber Bragg Grating*), por ejemplo.

En este caso se utilizará una configuración de la fibra óptica como sensor intrínseco, en el que los dos haces de luz del esquema de la **Figura 4** se sustituyen por dos brazos de fibra óptica: uno de referencia y otro de medida (del cual se pretende obtener el valor de la magnitud física). Cualquier variación que surja durante el camino óptico en cualquiera de ellos dará lugar un patrón de interferencia distinto, por lo que normalmente a través del brazo de referencia se compensan los efectos ambientales que puedan influir en la medida. No obstante, los puntos críticos en cuanto a pérdida de intensidad solo son los extremos finales de entrada y de salida de la fibra óptica, facilitando medidas remotas. La detección en el extremo final de salida de la fibra óptica se realiza mediante fotodetectores sensibles a la luz incidente sobre ellos

(normalmente después de colimarla por el efecto divergente a la salida), filtrando espacialmente el patrón de interferencia detectando únicamente la parte central y eliminando la necesidad de analizar un patrón de interferencia completo. En la **Figura 6** se representa el camino que recorre la luz en ambos brazos del interferómetro, indicando la intensidad óptica en cada punto de su longitud y la fase óptica detectada al final.

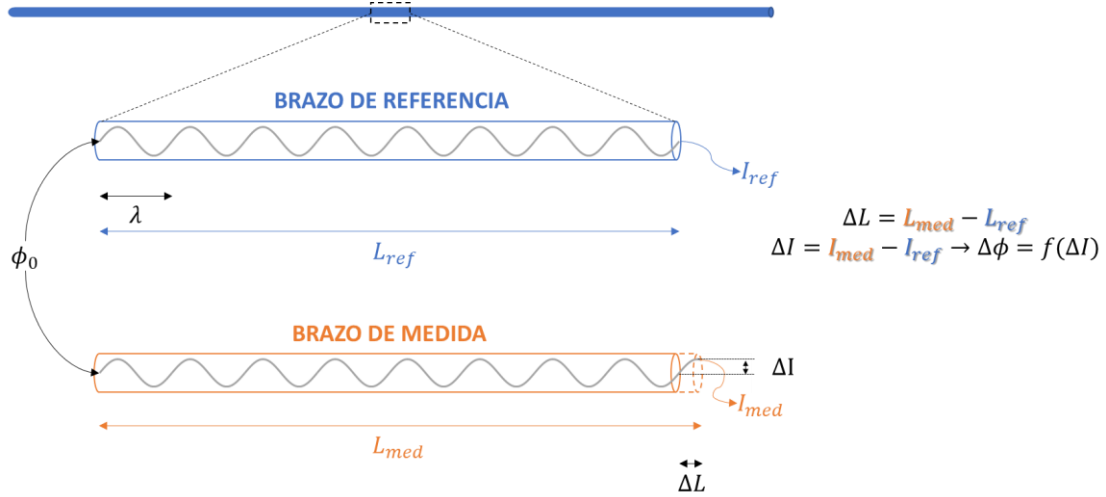


Figura 6. Diferencias en el camino óptico recorrido: intensidad y fase óptica detectadas

Una vez implementado el sensor de fibra óptica, es necesario conocer la relación entre la magnitud física a medir y la diferencia de fase óptica obtenida. Se parte de una fase óptica inicial ϕ_0 , que es la misma para ambos brazos del interferómetro puesto que procede de la misma fuente de luz. La fase varía a lo largo de la longitud de la fibra óptica L (misma longitud inicial en ambos brazos de fibra óptica), incrementándose de forma constante con la longitud recorrida por la luz en un factor $\frac{2\pi}{\lambda} n$ teniendo en cuenta que el índice de refracción n no varía a lo largo del camino. La diferencia de fase se obtiene finalmente relacionando ambas intensidades de luz obtenidas a partir de los fotodetectores en el extremo de cada brazo.

Puesto que el interferómetro se basa en fibra óptica con un índice de refracción n y no en espejos mediante luz directa en espacio libre, el camino óptico β se expresa como:

$$\beta = nL \quad (2.13)$$

Por tanto, la fase óptica depende tanto del índice de refracción como de la longitud física de la fibra óptica. Su diferencia se calcula entonces derivando la expresión (2.6) y sustituyendo el camino óptico por la relación (2.13):

$$\partial \phi = \left(\frac{2\pi}{\lambda} \right) \partial \beta = \left(\frac{2\pi}{\lambda} \right) (n \cdot \partial L + L \cdot \partial n) \quad (2.14)$$

Si el índice de refracción es constante (característica del núcleo de las fibras ópticas monomodo que se utilizarán en este proyecto), se obtiene una variación de la fase óptica dependiente únicamente de la longitud de la fibra óptica:

$$\partial\phi|_{n=cte} = \left(\frac{2\pi}{\lambda}n\right)\partial L \quad (2.15)$$

En resumen, el pequeño desplazamiento en el tiempo se obtiene en función de la variación de fase óptica existiendo una relación lineal:

$$L_{med}(t) - L_{ref} = \frac{\lambda}{2\pi n}(\phi_{med}(t) - \phi_{ref}) \quad (2.16)$$

donde $\phi_{med}(t)$ y ϕ_{ref} son las fases ópticas de medida y de referencia, y $L_{med}(t)$ y L_{ref} son las longitudes de los brazos de medida y de referencia de la fibra óptica.

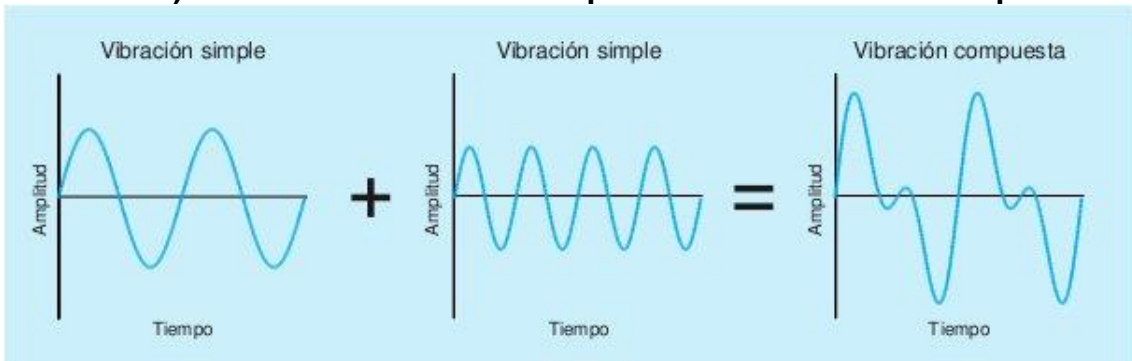
El trabajo realizado posibilitará, entre otras cosas, la medición de vibraciones de máquinas que también se manifiestan como una deformación en el material. Las microdeformaciones originadas en la fibra óptica se aprovechan para obtener la variación de la fase óptica según la siguiente expresión:

$$\frac{\partial\phi}{L} = \left(\frac{2\pi}{\lambda}n\right)\xi \cdot \varepsilon \quad (2.17)$$

donde $\varepsilon = \frac{\partial L}{L}$ es la deformación y ξ es un factor de corrección (debido al recubrimiento de la fibra óptica, la deformación lateral o el número de veces que la fibra óptica hace contacto con el objeto, por ejemplo)

Las vibraciones en general pueden ser simples (un solo armónico) o compuestas (superposición de dos o más armónicos), como se muestra en la **Figura 7**.

a) Suma de vibraciones simples en el dominio del tiempo



b) Suma de vibraciones simples en el dominio de la frecuencia

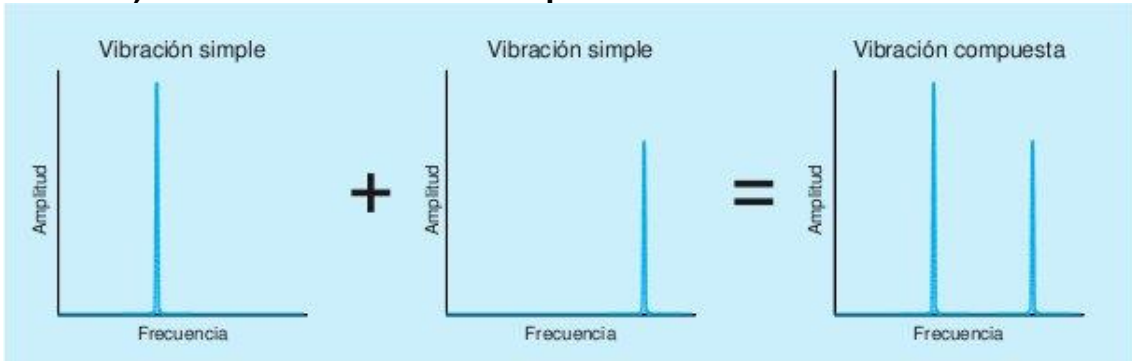


Figura 7. Vibraciones simples y compuestas [10]

Las vibraciones en máquinas rotatorias, por ejemplo, dan información sobre su correcto funcionamiento o la existencia de algún desajuste, como se puede ver en la **Figura 8**.

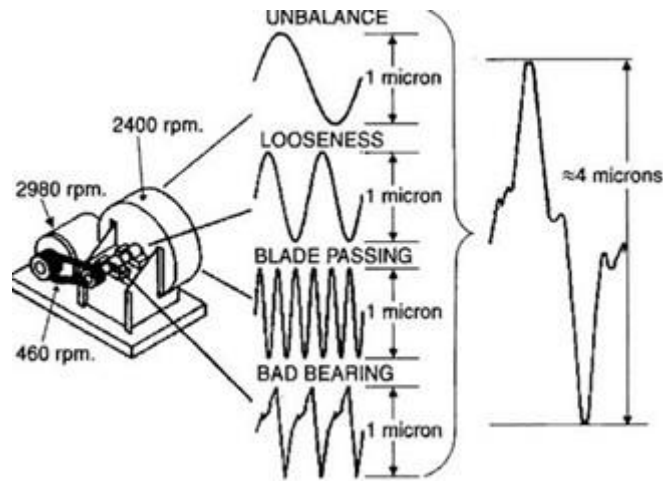


Figura 8. Vibración compuesta originada por los desajustes de una máquina rotatoria [11]

2.4. TÉCNICAS DE DETECCIÓN DE LA FASE ÓPTICA

La fase óptica a la salida del interferómetro podría obtenerse a priori resolviendo la ecuación (2.11). Sin embargo, se presentan varias dificultades siendo necesario aplicar un método específico para la detección de la misma. Existen diferentes y numerosas técnicas que buscan obtener la fase óptica con gran precisión basándose normalmente en añadir información a la señal de interferencia. Aunque existen varios criterios de clasificación de las técnicas de detección de fase óptica, principalmente se pueden englobar en dos grupos en función de las frecuencias utilizadas en cada uno de los haces del interferómetro [12]:

- **Detección homodina:** los dos haces de luz del interferómetro presentan la misma frecuencia, por lo que la ecuación (2.11) no varía:

$$I_{hom} = I_0(1 + V \cos(\Delta\phi)) \quad (2.18)$$

- **Detección heterodina:** los dos haces de luz del interferómetro presentan frecuencias diferentes. La ecuación (2.11) se transforma en:

$$I_{het} = I_0(1 + V \cos((\omega_2 - \omega_1)t + \Delta\phi)) \quad (2.19)$$

donde ω_2 y ω_1 son las frecuencias de los respectivos brazos del interferómetro.

A pesar de que la detección homodina presenta un alto rango dinámico y permite la detección a largas distancias (incluso kilómetros), la detección heterodina lleva consigo una serie de ventajas como la posibilidad de eliminar perturbaciones o la insensibilidad a la variación de intensidad de la luz. Además, con diferentes técnicas se han logrado mediante este tipo de detección una alta resolución y exactitud en la medida [13]. La modulación (en fase y/o amplitud) puede ser interna o externa:

- **Modulación interna:** también llamada modulación directa, se trata de la modulación directa del láser (modulando la corriente de inyección del láser modificando su longitud de onda).
- **Modulación externa:** precisa de moduladores externos entre los que destacan los electro-ópticos (EOM, *Electro-Optic Modulator*) en los que una señal eléctrica origina un cambio en el índice de refracción del material, y los acusto-ópticos (AOM, *Acousto-Optic Modulator*) en los que una onda acústica da lugar a un cambio periódico en el índice de refracción de la cavidad del modulador (obteniendo dos haces separados tanto físicamente como en frecuencia).

No obstante, en algunas de estas técnicas de modulación no se pueden aplicar las técnicas de detección de fase convencionales en métodos de desplazamiento de fase [14]. Sin embargo, las ventajas de un interferómetro heterodino pueden aplicarse a un interferómetro homodino si se aplica una tecnología de modulación concreta y se emplea un método de obtención de la fase que le corresponda, evitando la complejidad y el coste de las modulaciones anteriores. La modulación mediante un actuador piezoeléctrico (PZT), técnica ampliamente utilizada [13], es una buena solución ya que la estructura simple, la estabilidad, la relativamente alta velocidad de respuesta y el bajo coste hacen al PZT el candidato ideal para modificar la fase relativa entre ambos brazos del interferómetro.

En cuanto al método de procesamiento de la fase, las técnicas se pueden dividir en dos grupos: electrónicas y analíticas. Para determinar la fase electrónicamente se necesitan elementos específicos entre los que se incluyen detectores de cruce por cero, lazos de seguimiento de fase (PLL, *Phase-Locked Loops*) o contadores que ayuden a monitorizar la intensidad de salida de la fibra (o del interferograma) mientras la fase es modulada. Para determinar la fase analíticamente, los datos sobre la intensidad se guardan mientras la fase es temporalmente modulada, enviados al ordenador, y después utilizados para calcularla.

En el trabajo se desarrollará una técnica de detección síncrona de la fase óptica con la que se pretende obtener varias ventajas. Para ello, se obtendrán dos señales en cuadratura a la salida del interferómetro a través del muestreo síncrono en los instantes correspondientes modulando con el PZT, y a partir de ambas señales se demodulará la fase óptica analíticamente. Aunque se aprovechará la modulación mediante el PZT, existen otros métodos para obtener dos señales en cuadratura modulando el láser o mediante elementos ópticos [15].

Una técnica similar en los fundamentos en los que se basa la presentada en este trabajo es la técnica de la portadora generada en fase (PGC, *Phase Generated Carrier*), en la que se utilizan dos señales a diferentes frecuencias para obtener la fase óptica [16]. Es uno de los algoritmos de demodulación más populares en sistemas de fibra óptica por su alta resolución y por no renunciar a la alta sensibilidad, aunque precisa de cálculos complejos y por tanto de un sistema de demodulación sofisticado de alto rendimiento.

En resumen, en este trabajo se utilizará un interferómetro de Mach-Zehnder en una configuración completa de fibra óptica que se comportará como sensor intrínseco. A continuación, se desarrollará una técnica de detección síncrona de la fase óptica y después se integrará en un sistema completo para la medición de vibraciones u otras magnitudes.

3. DESARROLLO DE LA TÉCNICA DE DETECCIÓN SÍNCRONA DE LA FASE ÓPTICA

3.1. SISTEMA ÓPTICO Y OBTENCIÓN DE LA FASE ÓPTICA

En este proyecto se va a diseñar e implementar una técnica de detección síncrona de la fase óptica en un interferómetro Mach-Zehnder basado en fibra óptica, obteniendo una serie de ventajas frente a otras técnicas utilizadas.

La técnica consiste, como primera aproximación, en la modulación de la fase óptica mediante una frecuencia alta o frecuencia de la señal de modulación que posteriormente permitirá extraer la información de frecuencias más bajas objeto de medida (frecuencias de vibración, por ejemplo), demodulando la fase óptica a la salida en instantes concretos. El esquema óptico del interferómetro utilizado se muestra en la **Figura 9**.

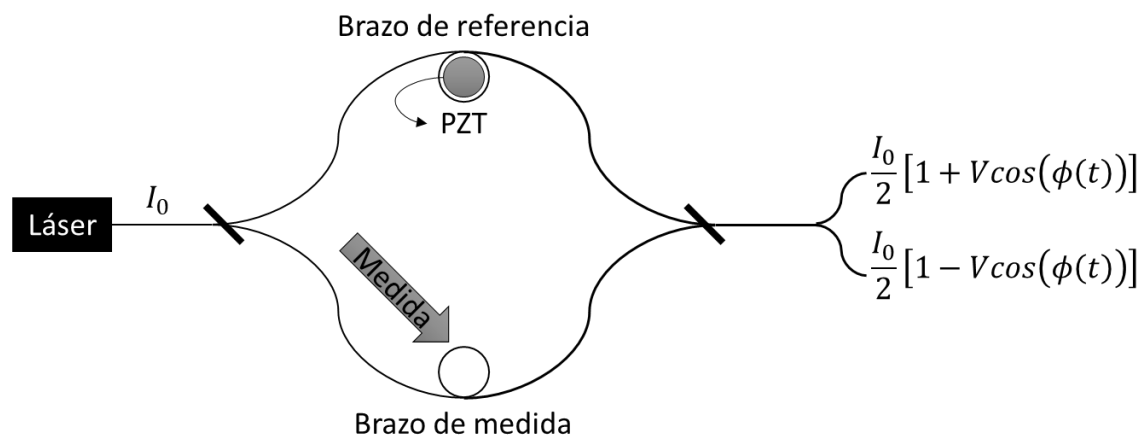


Figura 9. Esquema óptico del interferómetro Mach-Zehnder basado en fibra óptica

La introducción de la señal de modulación en el sistema se realiza excitando con cierta tensión variable el PZT, que modula la fase óptica al deformar la fibra óptica (brazo de referencia) cambiando la forma según la tensión aplicada (propiedad básica de un material piezoeléctrico). De esta manera, las variaciones en la fase óptica del brazo de medida podrán ser detectadas una vez demodulada la fase óptica a la salida del interferómetro, que responde a la función de transferencia de la ecuación (2.11) como se indica en la **Figura 9**.

La instalación en el laboratorio se ilustra en la **Figura 10**. Consta principalmente de un láser (proporciona la potencia óptica), un posicionador de tres ejes con una lente de microscopio (para dirigir la intensidad óptica del láser a la fibra óptica), un separador de haz (acoplador de fibra óptica que divide la potencia óptica del láser en dos haces), dos brazos de fibra óptica (uno de medida y el otro de referencia por donde circulan los dos haces) y un recombinador (acoplador de fibra óptica que recombina ambos haces para obtener la interferencia entre ambos haces). Una parte de la fibra óptica del brazo de referencia se enrolla alrededor del PZT

y las dos salidas del recombinador (cada una con un término constante y el otro contrario en signo) se conectan a dos fotodiodos con un amplificador de transimpedancia (fotodiodos amplificados balanceados). Además, se ha añadido un control de polarización para que los dos haces se encuentren igualmente polarizados y se produzca la interferencia entre ambos.



Figura 10. Instalación del interferómetro (sistema óptico) en el Laboratorio de Optoelectrónica (UC3M)

Concretamente, el láser utilizado es monocromático de He-Ne con una longitud de onda $\lambda = 633 \text{ nm}$ (rojo) y una potencia $P = 1 \text{ mW}$ (alimentado con la fuente de alimentación indicada), y la fibra óptica monomodo presenta un índice de refracción $n = 1.46$. La importancia de estos dos elementos radica en que la luz del láser debe ser muy colimada, estable y con bajo ruido en fase, y la fibra óptica monomodo debe ser compatible con la longitud de onda del mismo. A partir de este montaje de referencia se puede desarrollar la técnica de detección síncrona para obtener la fase óptica objeto de medida.

La base de la técnica es la modulación de la fase óptica mediante una señal de tal manera que se obtengan a la salida del interferómetro dos señales en cuadratura, o lo que es lo mismo, la respuesta del interferómetro para dos fases ópticas distintas separadas $\frac{\pi}{2}$ radianes (que también podrían obtenerse a la vez en diferentes puntos espaciales de un patrón de interferencia). Esto puede lograrse a partir del muestreo síncrono a la salida en los instantes en los que ocurren los máximos y mínimos de una señal de modulación de la fase óptica sinusoidal con amplitud de pico a pico igual a $\frac{\pi}{2}$ radianes. Analizando la fase óptica total que se obtendría a la salida del interferómetro, esta se puede desglosar en la suma de tres contribuciones (despreciando las variaciones de fase originadas por factores externos no deseados: perturbaciones y la deriva de la fase óptica por la temperatura, por ejemplo):

$$\phi(t) = \phi_0 + \phi_{med}(t) + \phi_{mod}(t) \quad (3.1)$$

donde ϕ_0 es la fase óptica inicial, $\phi_{med}(t)$ es la fase óptica de medida y $\phi_{mod}(t)$ es la fase óptica de modulación.

La deriva de la fase óptica por la temperatura (aumento constante de la fase óptica) quizás sea una de las influencias más importantes que pueden dificultar la obtención de la fase óptica. Sin embargo, esta deriva es de muy baja velocidad, por lo que igualmente se permiten realizar mediciones de magnitudes de muy baja frecuencia (incluso por debajo de los 5 Hz sería aceptable).

La fase óptica de modulación $\phi_{mod}(t)$ viene dada por la frecuencia de la señal de modulación sinusoidal con la que se excita el PZT (f_{mod}):

$$\phi_{mod}(t) = \phi_{mod0} \cdot \text{sen}(\omega_{mod}t) = \phi_{mod0} \cdot \text{sen}(2\pi f_{mod}t) \quad (3.2)$$

donde ϕ_{mod0} es la amplitud de pico de la fase óptica de modulación y ω_{mod} es la frecuencia angular de la señal de modulación.

Para obtener las dos señales en cuadratura es por tanto necesaria una amplitud de pico $\phi_{mod0} = \frac{\pi}{4}$ radianes. En cuanto a la salida, muestrearla en los instantes en los que se produce un máximo o un mínimo en la señal de modulación para obtener dichas señales es equivalente a muestrear en los instantes de tiempo k_M (para los máximos) en los que $\omega_{mod}t = \frac{\pi}{2} + 2\pi n$, y k_m (para los mínimos) en los que $\omega_{mod}t = \frac{3\pi}{2} + 2\pi n$ (con $n = 0, 1, 2 \dots$).

Así se obtienen las dos señales en cuadratura $I_M(k)$ y $I_m(k)$, cuyos valores responden a la función de transferencia del interferómetro reescrita en tiempo discreto (k):

$$\begin{aligned} I(k) &= \frac{I_0}{2} (1 + V \cdot \cos(\phi_0 + \phi_{med}(k) + \phi_{mod}(k))) \\ \begin{cases} I_M(k_M) = \frac{I_0}{2} (1 + V \cdot \cos(\phi_0 + \phi_{med}(k_M) + \phi_{mod0})) \\ I_m(k_m) = \frac{I_0}{2} (1 + V \cdot \cos(\phi_0 + \phi_{med}(k_m) - \phi_{mod0})) \end{cases} \end{aligned} \quad (3.3)$$

donde I_0 es la intensidad óptica proporcionada por el láser y V es la visibilidad del interferómetro.

Las dos respuestas del interferómetro para dos fases ópticas desplazadas $\frac{\pi}{2}$ radianes (siendo equivalente a obtener la misma función de transferencia, pero cambiando el coseno por el seno) se corresponden a una fase óptica concreta como se puede observar en la **Figura 11**. Se observa que la respuesta no presenta una componente continua. Esto se debe al uso posterior de los fotodiodos y el amplificador de transimpedancia obteniéndose la diferencia entre las dos señales ópticas de las dos salidas del recombinador. De esta manera, la intensidad óptica de salida final I resulta ser el valor duplicado del término que incluye la información de la fase óptica (término de interés):

$$I = \frac{I_0}{2} [1 + V \cdot \cos(\phi(t))] - \frac{I_0}{2} [1 - V \cdot \cos(\phi(t))] = I_0 V \cdot \cos(\phi(t)) \quad (3.4)$$

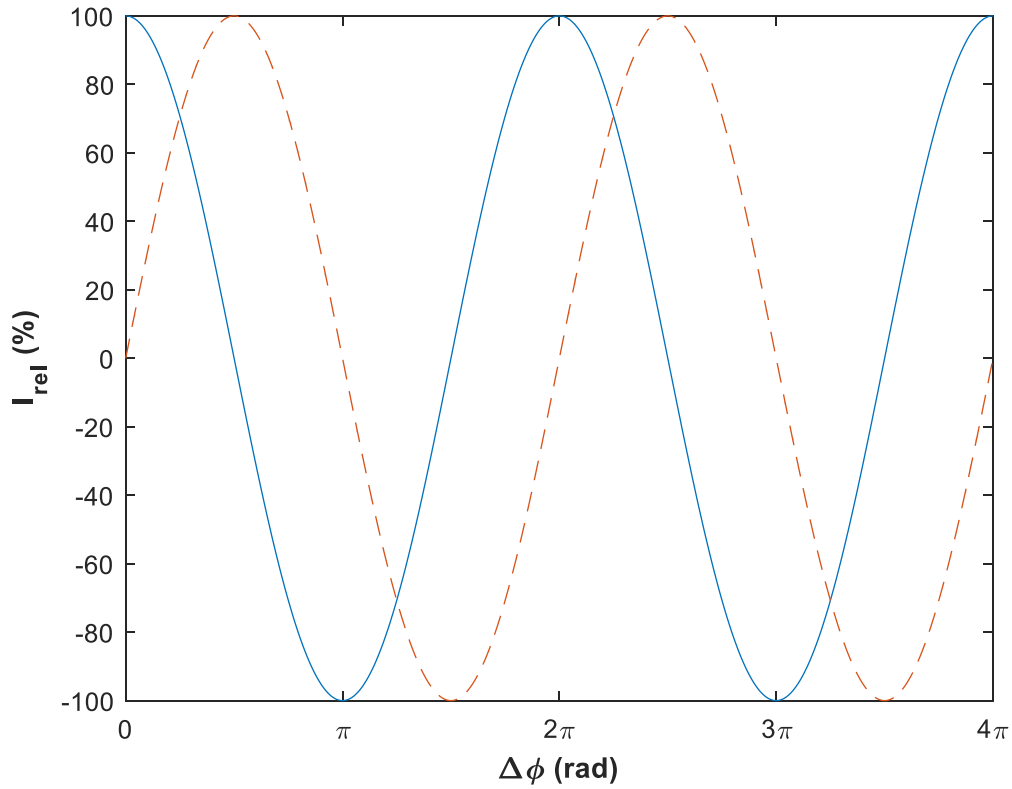


Figura 11. Función de transferencia del interferómetro original (azul) y la misma desfasada $\pi/2$ radianes (naranja) sin componente continua (respuesta correspondiente a las dos señales en cuadratura)

Una vez obtenidas las muestras de ambas señales se procede a la demodulación de la fase óptica. Conociendo que una de las señales en cuadratura es el coseno y la otra es el seno, ambas sin componente continua y multiplicadas por la intensidad óptica del láser y la visibilidad del interferómetro, se puede obtener la tangente correspondiente a la fase óptica en cada instante muestreado k' (correspondiente al instante de la mitad del intervalo de tiempo entre máximo y mínimo consecutivos).

$$\tan \phi(k') = \frac{I_0 V \cdot \sin \phi(k')}{I_0 V \cdot \cos \phi(k')} = \frac{\sin \phi(k')}{\cos \phi(k')} \quad (3.5)$$

La fase óptica instantánea $\phi_i(k)$ se obtiene por tanto aplicando la función arcotangente al resultado anterior, obteniéndose una muestra por cada dos muestras de la señal de salida (una de cada señal en cuadratura) y pudiendo compensarse opcionalmente restando $\frac{\pi}{4}$ radianes debido a la modulación en fase óptica de $\pm \frac{\pi}{4}$ radianes. La fase óptica acumulada $\phi_a(k)$ se obtiene sumando las variaciones de fase entre las fases ópticas instantáneas consecutivas, siendo ϕ_{a_0} la fase óptica acumulada inicial:

$$\begin{aligned} \phi_i(k) &= \arctan(\tan \phi(k)) \\ \phi_a(k) &= \phi_{a_0} + \sum_{k=0}^{k-1} (\phi_i(k+1) - \phi(k)) \end{aligned} \quad (3.6)$$

Aunque mediante este método se consigue la determinación exacta de la fase, existen puntos críticos a tener en cuenta. Estos puntos son aquellos en los que la tangente tiene un valor infinito, y por tanto la fase óptica vale $\pm \frac{\pi}{2}$ radianes. En estos puntos se produce una discontinuidad de $\frac{\pi}{2}$ a $-\frac{\pi}{2}$ o al revés, y por tanto se deben considerar para acumular el desplazamiento de fase óptica correctamente y obtener así una curva continua y determinada para cualquier instante de tiempo. Para ello se tiene en cuenta la variación máxima existente en valor absoluto entre los dos puntos de alrededor (A y B), es decir, π radianes, y se calcula la diferencia de fase de ajuste necesaria ($\Delta\phi_{AB(\frac{\pi}{2} \rightarrow -\frac{\pi}{2})}$ o $\Delta\phi_{AB(-\frac{\pi}{2} \rightarrow \frac{\pi}{2})}$) para que la suma de esta y la diferencia de fase instantánea ($\phi_A - \phi_B$) sea $\pm\pi$ radianes según corresponda:

$$\begin{aligned}\Delta\phi_{AB(\frac{\pi}{2} \rightarrow -\frac{\pi}{2})} &= \pi + (\phi_B - \phi_A) \\ \Delta\phi_{AB(-\frac{\pi}{2} \rightarrow \frac{\pi}{2})} &= -\pi + (\phi_B - \phi_A)\end{aligned}\tag{3.7}$$

En definitiva, mediante esta técnica se obtienen cuatro ventajas principales en la determinación de la fase óptica frente a otras técnicas utilizadas:

- **Sensibilidad:** la sensibilidad varía en función del punto de trabajo del interferómetro debido al carácter no lineal y periódico de su función de transferencia. En este caso, cuando una de las señales en cuadratura se encuentra en uno de los extremos donde las funciones seno y coseno pierden sensibilidad, la otra se encuentra en la zona de mayor sensibilidad, resolviendo el problema.
- **Ambigüedad de signo:** se soluciona el problema de la ambigüedad del signo, ya que a partir de las funciones seno y coseno obtenidas se puede obtener la fase de forma inequívoca en un rango mayor y también el sentido de desplazamiento (determinar si se trata de un incremento o un decremento en la fase óptica con la información de la segunda señal cuando la primera se encuentra en el límite superior o inferior de la función de transferencia). Además, los cálculos realizados son relativamente sencillos.
- **Independencia de la amplitud:** una ventaja importante es la independencia de la amplitud de la intensidad óptica a la salida del interferómetro, así como de la visibilidad del interferómetro, ya que en este caso no son parámetros necesarios para el cálculo de la fase óptica. Además, la responsividad de los fotodetectores depende de la longitud de onda del láser y por tanto sería necesario conocer las características del láser y de los fotodetectores para su determinación. Debido a la independencia de la amplitud, también se reduce la influencia de la birrefringencia de la fibra óptica (efecto que da lugar a un desvanecimiento de la señal al cambiar la polarización a la salida debido a la variación de las propiedades de la fibra óptica por torsión, temperatura, etc.).
- **Punto de trabajo variable:** mediante este método no es necesario mantener el interferómetro en un punto de trabajo fijo como se realiza en otro tipo de técnicas mediante PLL, por ejemplo. En estas últimas se busca el punto en el que la sensibilidad es máxima y la función de transferencia es lineal ($\phi = \frac{\pi}{2}$).

En el siguiente apartado se demuestra la validez de la técnica desarrollada.

3.2. SIMULACIÓN Y PROCESAMIENTO DE LOS RESULTADOS

Una vez diseñada la técnica de detección de la fase óptica, se puede probar su funcionamiento simulando el comportamiento del interferómetro. Para ello se ha utilizado el programa MATLAB¹. En este programa se han incluido las ecuaciones necesarias para simular el comportamiento que se obtendría idealmente en el interferómetro, además de añadirse el algoritmo de demodulación de la fase óptica para obtener finalmente los resultados de aplicar el mismo. El código fuente desarrollado se encuentra en el **ANEXO I: Algoritmo de detección síncrona de la fase óptica**.

Como señal de modulación (V_i) del sistema, es decir, la señal aplicada al PZT, se va a utilizar una señal sinusoidal de 20 kHz tanto en la simulación como en el sistema real, evitando el rango de frecuencias en las que se pueda manifestar el efecto de resonancia del mismo. Aunque para el fin deseado en el que se adquiere únicamente en los máximos y mínimos cabe pensar que podría utilizarse una señal cuadrada, esto es cierto en una simulación (el algoritmo se ha preparado para detectar los máximos y mínimos de cualquier tipo de señal: triangular, cuadrada, sinusoidal, etc.) pero no es conveniente para el interferómetro real debido a las características del PZT. Este filtraría la mayoría de las componentes de la señal comportándose como un filtro paso bajo, siendo por tanto mejor aplicar directamente una señal sinusoidal.

La señal debe proporcionar un desfase $\Delta\phi_{pp}$ de $\frac{\pi}{2}$ radianes en la fase óptica entre su máximo y su mínimo, o bien cumplir la siguiente relación (obteniéndose el mismo resultado: dos señales en cuadratura):

$$\Delta\phi_{pp} = \frac{\pi}{2} + 2\pi n \quad n = 0, 1, 2, \dots \quad (3.8)$$

En este caso simulado se ha escogido una amplitud de pico de la fase de modulación $\phi_{mod_0} = \frac{\pi}{4}$ radianes, que resulta en un desfase entre máximo y mínimo consecutivos $\Delta\phi_{pp} = \frac{\pi}{2}$ radianes. En la **Figura 12** se representa la señal de modulación relativa que se aplicaría al PZT (V_{irel}) para lograr la modulación en fase óptica deseada, con los máximos y mínimos marcados (instantes en los que se muestrea la señal de salida del interferómetro).

La tensión variable aplicada al PZT que actúa como señal de modulación debe ajustarse de tal manera que en la salida del interferómetro se observe una diferencia de fase óptica entre máximo y mínimo consecutivos que cumpla la relación (3.8). Sin embargo, en ocasiones puede resultar complicado realizar esta calibración manualmente. Por ello, en el sistema completo se recurrirá a una calibración automática a través de un microcontrolador para librarse de esta tarea y realizarla de forma más rápida y sencilla.

¹ Más información: <https://es.mathworks.com/products/matlab.html>

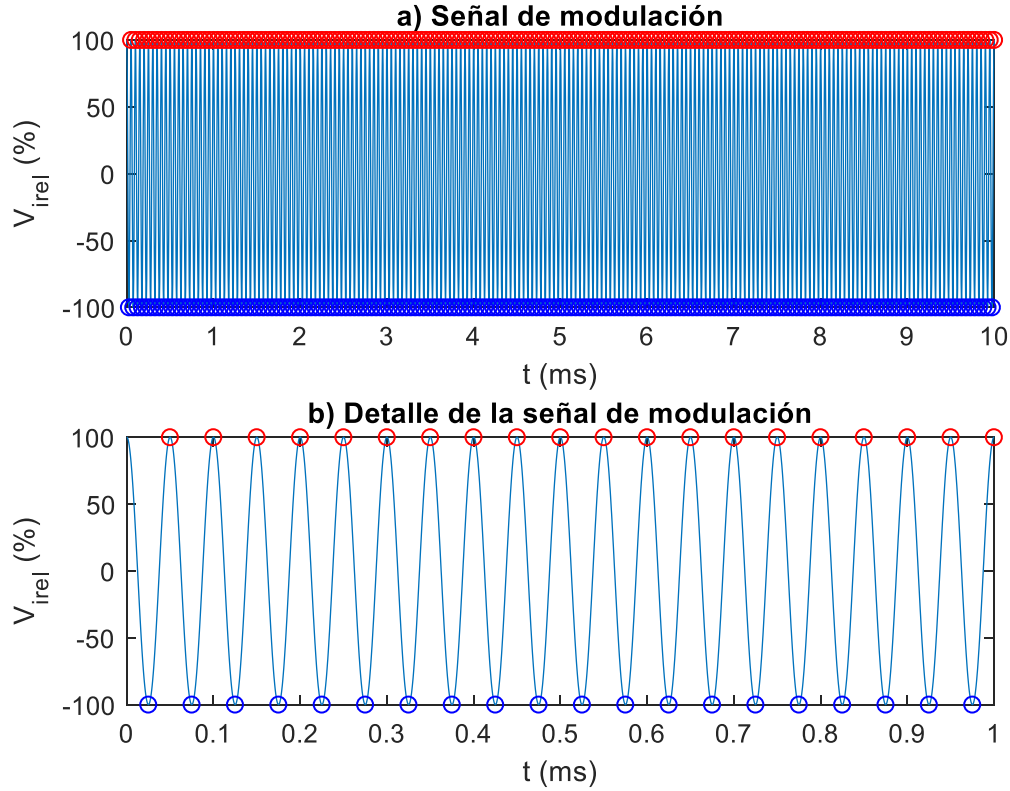


Figura 12. Señal de modulación relativa (modulación de la fase óptica) con los máximos (rojo) y mínimos (azul) marcados

Como señal de medida, primero se va a suponer una única deformación sinusoidal en el tiempo con una frecuencia $f_{med} = 100 \text{ Hz}$ y con una amplitud de pico de la fase óptica de medida $\phi_{med_0} = 5 \text{ rad}$ (superando los límites superior e inferior de $\pm \frac{\pi}{2}$ radianes de la función arcotangente):

$$\phi_{med}(t) = \phi_{med_0} \text{sen}(2\pi f_{med}t) \quad (3.9)$$

$$\phi_{med}(t) = 5 \cdot \text{sen}(200\pi t) \text{ rad}$$

A la salida del interferómetro se obtendrá una intensidad en respuesta a esta señal que en el montaje será detectada por los fotodetectores amplificados balanceados que además eliminarán la componente continua no deseada. De estos fotodetectores se obtiene una señal de tensión eléctrica que puede ser visualizada en un osciloscopio y cuya amplitud varía en función de la intensidad óptica proporcionada por el láser y de la visibilidad del interferómetro, entre otros parámetros. En la simulación se obtiene la señal de salida relativa que se obtendría en el montaje real. La señal de salida en este caso está representada en la **Figura 13**, donde se encuentran las dos señales en cuadratura muestreadas y correspondientes a los instantes en los que se produce un máximo (rojo) o un mínimo (azul) en la señal de modulación.

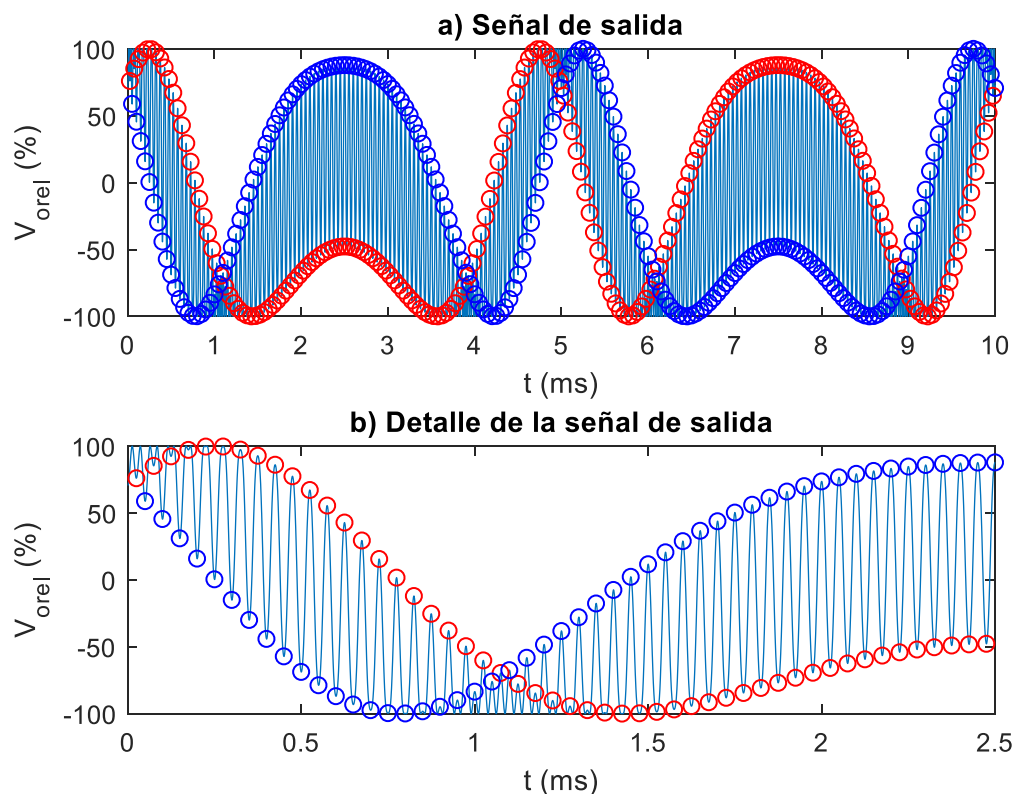


Figura 13. Señal de salida relativa (salida del interferómetro después de los fotodiodos amplificados balanceados) con las dos señales en cuadratura marcadas (rojo y azul)

Una vez obtenidas las muestras de ambas señales en cuadratura, se aplican las relaciones obtenidas en las ecuaciones del apartado anterior para demodular la fase óptica instantánea. No obstante, además de obtener la tangente a partir de estas señales también se va a aprovechar la información individual de cada una de ellas para duplicar el rango de fase instantánea disponible. Si se conoce el signo de las dos señales (equivalentes al seno y al coseno), se puede determinar la fase instantánea en un rango de $[-\pi, \pi]$ radianes en lugar de en un rango de $[-\frac{\pi}{2}, \frac{\pi}{2}]$ radianes que se dispone aplicando exclusivamente la función arcotangente, como se puede observar en la **Tabla 1**.

Tabla 1. Relación entre los signos de las señales en cuadratura, los rangos de fase y la fase ajustada

$\sin \phi$	$\cos \phi$	$\tan \phi$	Rango ϕ (rad)	ϕ Ajustada (rad)
+	+	+	$[0, \frac{\pi}{2}]$	$+\lvert\phi\rvert$
+	-	-	$[\frac{\pi}{2}, \pi]$	$\pi - \lvert\phi\rvert$
-	+	-	$[-\frac{\pi}{2}, 0]$	$-\lvert\phi\rvert$
-	-	+	$[-\pi, -\frac{\pi}{2}]$	$\pi + \lvert\phi\rvert$

Para lograr esto se hace uso de una función específica de MATLAB para calcular el arcotangente: “*atan2(Y, X)*”. De esta manera, proporcionando los valores X (coseno) e Y (seno) del cociente Y/X (tangente) se obtiene directamente la fase instantánea en el rango $[-\pi, \pi]$ radianes.

La fase óptica instantánea y la fase óptica acumulada, que contiene la suma de las variaciones de la fase óptica instantánea con las discontinuidades corregidas, se representan en la **Figura 14**. El desplazamiento equivalente debido a la deformación se indica en la escala de la derecha, obtenido a partir de la ecuación (2.16).

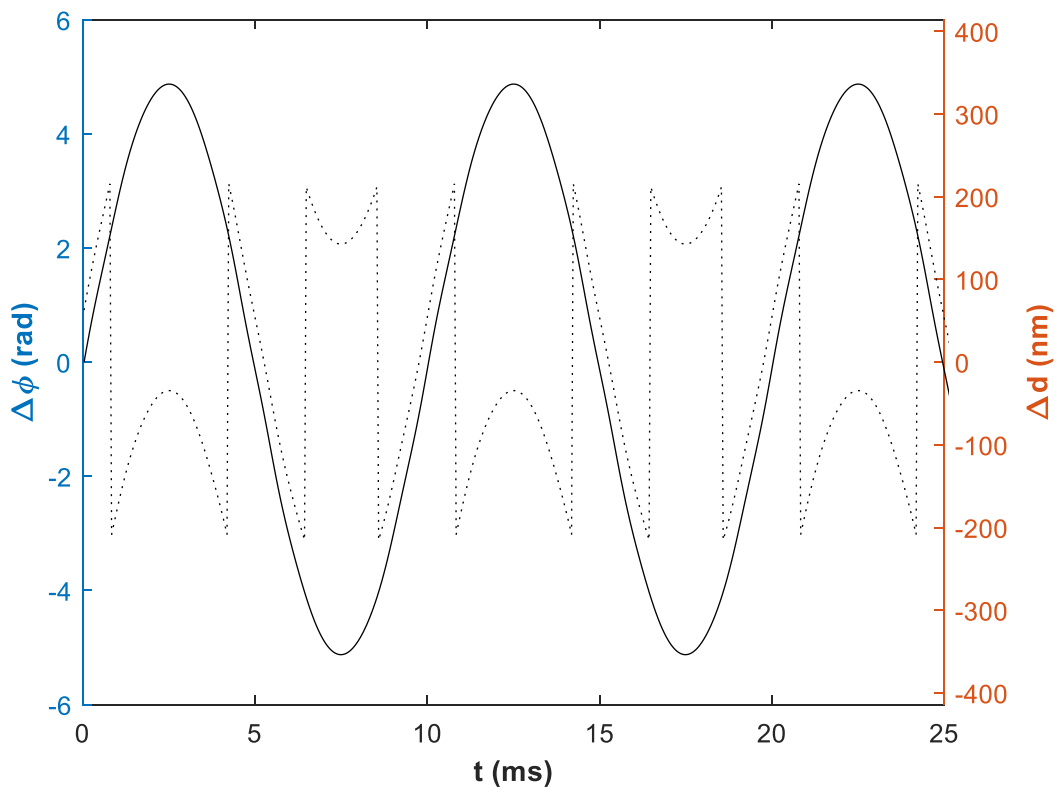


Figura 14. Fase óptica demodulada: instantánea (línea punteada) y acumulada (línea continua), con el desplazamiento equivalente a la derecha

Hasta ahora se ha obtenido la evolución temporal de la fase óptica de medida. Sin embargo, para el estudio de ciertas magnitudes físicas y en concreto de las vibraciones es más práctico conocer el espectro de frecuencias de la medida obtenida que dé a conocer las componentes de la misma. Para ello se ha calculado la DFT (*Discrete Fourier Transform*) mediante FFT² (*Fast Fourier Transform*). El algoritmo de cálculo de la DFT funciona de forma óptima (más rápido y más preciso) cuando las muestras N proporcionadas son potencia de dos: $N = 2^n$ ($n \in \mathbb{N}$). En este caso se han utilizado $2^{13} = 8192$ muestras (fase óptica acumulada) y el espectro de frecuencias resultante se muestra en la **Figura 15**.

² Las funciones FFT de MATLAB están basadas en la librería FFTW. Más información: <http://www.fftw.org>

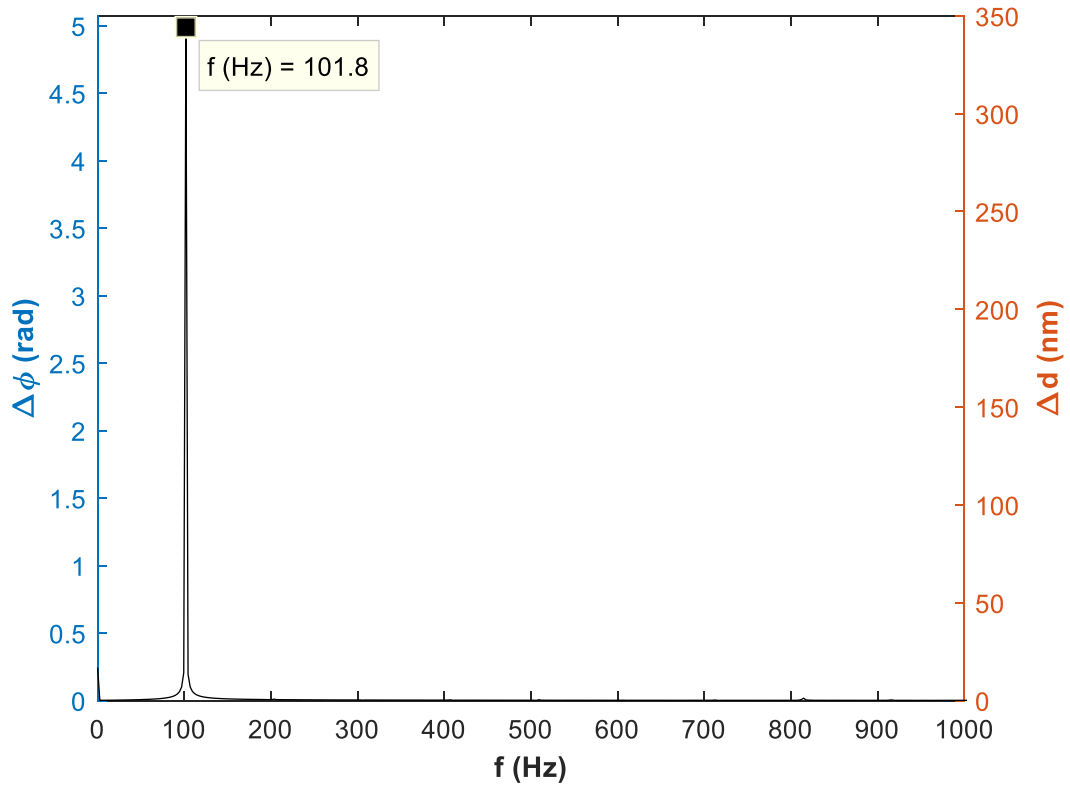


Figura 15. Espectro de frecuencias de la fase óptica acumulada, con el desplazamiento equivalente a la derecha

De esta manera, a partir del resultado final del espectro de frecuencias se puede determinar la amplitud de la fase óptica de medida y determinar las vibraciones que se producen en el brazo de medida del interferómetro para realizar el diagnóstico de una máquina, por ejemplo. El caso anterior se corresponde a una vibración simple, de un solo armónico. Sin embargo, las vibraciones también pueden ser compuestas consistiendo en la suma de diferentes armónicos. Para demostrar la efectividad de la técnica en la detección de este tipo de vibraciones, en la **Figura 16** se muestra el resultado final obtenido para una vibración compuesta de tres contribuciones (armónicos) cuya fase óptica $\phi_{med}(t)$ varía en el tiempo de la siguiente forma:

$$\begin{aligned} \phi_{med}(t) = & \phi_{0(100 \text{ Hz})} \text{sen}(2\pi \cdot 100t) + \phi_{0(200 \text{ Hz})} \text{sen}(2\pi \cdot 200t) \\ & + \phi_{0(500 \text{ Hz})} \text{sen}(2\pi \cdot 500t) \end{aligned} \quad (3.10)$$

$$\phi_{med}(t) = 2 \cdot \text{sen}(200\pi t) + 1 \cdot \text{sen}(400\pi t) + 4 \cdot \text{sen}(1000\pi t) \text{ rad}$$

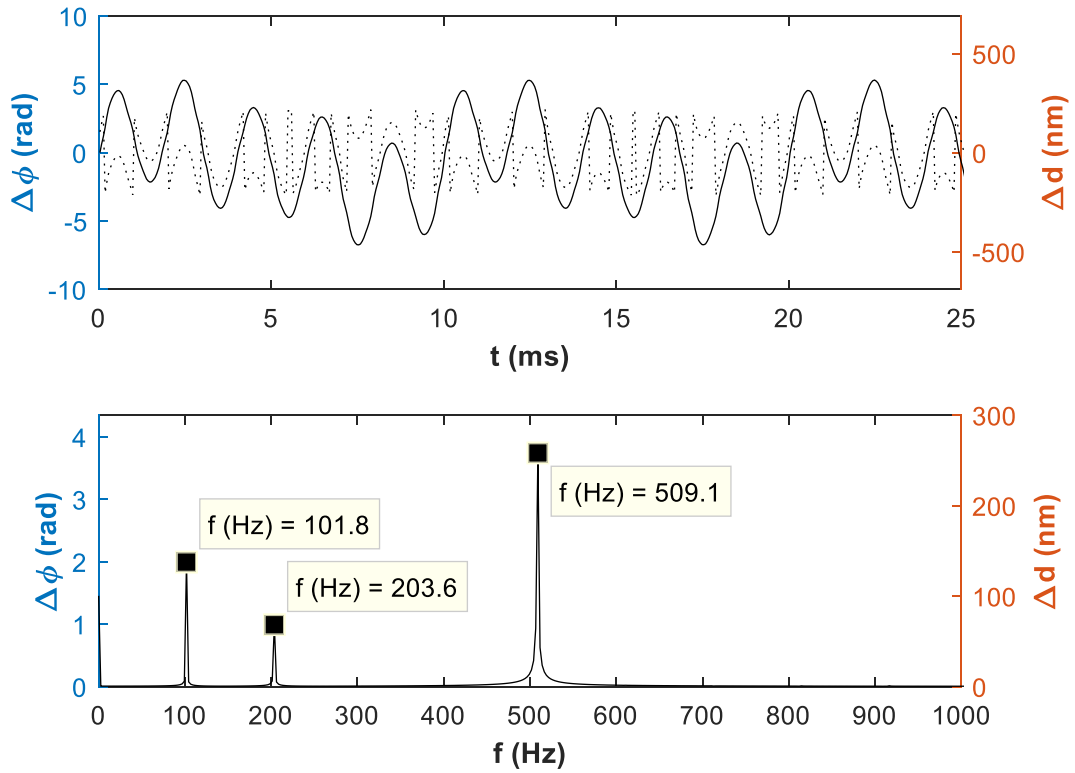


Figura 16. Resultados de aplicar la técnica de detección síncrona a una fase óptica de medida correspondiente a una vibración compuesta

Como se puede observar, los resultados obtenidos en ambos casos son coherentes con la magnitud de medida simulada de prueba. Más adelante, en el ejemplo de aplicación (Capítulo 5), se probará realmente la técnica desarrollada con la detección de las vibraciones en el núcleo de un transformador.

3.3. CONSIDERACIONES

Como en cualquier otra técnica, deben tenerse en cuenta las limitaciones a la hora de utilizarla. En este caso se identifican principalmente dos cuestiones a tener en cuenta antes de emplearla:

- **Frecuencia de muestreo:** la frecuencia de muestreo debe ser mayor que un cierto umbral para reconstruir la fase óptica acumulada correctamente.
- **Señales en cuadratura desfasadas:** existe cierto error al utilizar las dos señales en cuadratura al adquirirse en diferentes instantes de tiempo.

Comenzando con la frecuencia de muestreo, en la **Figura 17** se representa la fase óptica detectada tomando un número de muestras N diferente para muestrear una fase que varía linealmente de 0 a π radianes, teniendo en cuenta el salto de $\frac{\pi}{2}$ a $-\frac{\pi}{2}$ (debido a los límites de fase de la función arcotangente).

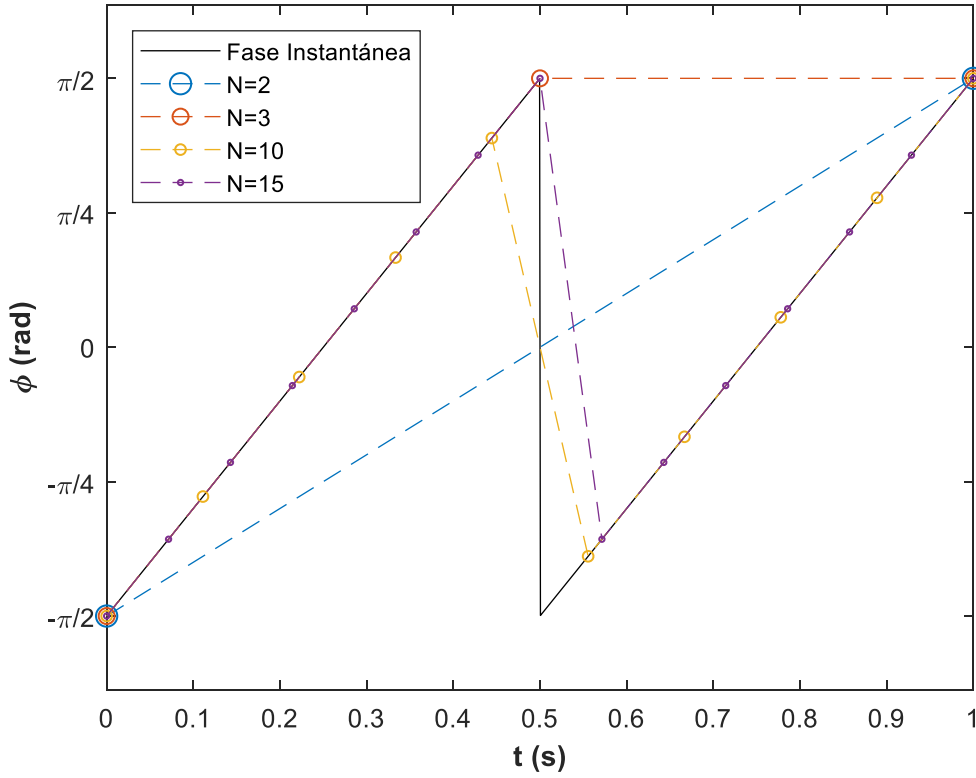


Figura 17. Fase óptica detectada muestreando con diferentes números de muestras (N) la fase óptica instantánea

De la **Figura 17** se concluye que la fase óptica debe ser muestreada a una frecuencia que permita detectar variaciones menores de $\frac{\pi}{2}$ radianes. En este caso, la condición se cumple para $N = \{10, 15\}$. De lo contrario, la reconstrucción la fase óptica total acumulada no se realizará correctamente, ya que esta se basa en los saltos mayores de $\frac{\pi}{2}$ radianes (o π radianes si se aplican las reglas de signo de la **Tabla 1**) para detectar las discontinuidades y unir los diferentes tramos, existiendo entonces una ambigüedad en la presencia o no de una discontinuidad pudiendo pasar inadvertida. De esta manera, la frecuencia de modulación mínima dependerá tanto de la amplitud máxima de la fase óptica objeto de medida (ϕ_{med_MAX}) como de la frecuencia del armónico de mayor frecuencia de la misma (f_{med_MAX}), siendo el peor caso:

$$\phi_{med}(t) = \phi_{med_MAX} \sin(2\pi f_{med_MAX} t) \quad (3.11)$$

Derivando la expresión anterior y obteniendo su valor para $t = 0$ se obtiene la pendiente máxima posible ($\frac{d\phi_{med}}{dt}_{MAX}$):

$$\frac{d\phi_{med}}{dt}_{MAX} = \left. \frac{d\phi_{med}}{dt} \right|_{t=0} = 2\pi f_{med_MAX} \phi_{med_MAX} \quad (3.12)$$

A partir de la pendiente máxima se obtiene la variación máxima permitida entre dos fases instantáneas consecutivas ($\Delta\phi_{med_MAX}$) y a partir de esta la frecuencia de modulación (f_{mod}) mínima para reconstruir satisfactoriamente la curva deseada:

$$\Delta\phi_{med_{MAX}} = 2\pi f_{med_{MAX}} \phi_{med_{MAX}} \cdot T_{mod} = \frac{2\pi f_{med_{MAX}} \phi_{med_{MAX}}}{f_{mod}} \quad (3.13)$$

$$\Delta\phi_{med_{MAX}} < \frac{\pi}{2} \rightarrow f_{mod} > 4f_{med_{MAX}} \phi_{med_{MAX}}$$

Cabe destacar que la frecuencia de modulación mínima puede reducirse a la mitad si se aplican las reglas de signo explicadas en el apartado anterior para duplicar el rango de fase instantánea disponible.

El segundo factor importante es el error generado al muestrear en los máximos y mínimos para obtener las dos señales en cuadratura, ya que existe una diferencia de tiempo en el que los parámetros seno y coseno adquiridos varían. El error máximo ocurre, de igual manera que en el análisis anterior, cuando la pendiente de la fase óptica objeto de medida es máxima. Considerando que la fase óptica instantánea calculada (ϕ') corresponde al instante de tiempo t_m en la mitad del intervalo $\Delta t = t_1 - t_2$ entre los instantes en los que se adquiere el seno y el coseno (t_1 y t_2), la tangente calculada ($\tan \phi'$) en función de la tangente real o esperada ($\tan \phi$) es:

$$t_2 = t_1 + \frac{T_{mod}}{2} = t_1 + \frac{1}{2f_{mod}}$$

$$\tan \phi' = \frac{\sin(\phi(t_2))}{\cos(\phi(t_1))} = \frac{\sin\left(\phi\left(t_m + \frac{\Delta t}{2}\right)\right)}{\cos\left(\phi\left(t_m - \frac{\Delta t}{2}\right)\right)} = \frac{\sin\left(\phi + \frac{2\pi f_{med_{MAX}} \phi_{med_{MAX}}}{4f_{mod}}\right)}{\cos\left(\phi - \frac{2\pi f_{med_{MAX}} \phi_{med_{MAX}}}{4f_{mod}}\right)} \quad (3.14)$$

$$= \frac{\sin \phi \cdot \cos\left(\frac{\pi f_{med_{MAX}} \phi_{med_{MAX}}}{2f_{mod}}\right) + \cos \phi \cdot \sin\left(\frac{\pi f_{med_{MAX}} \phi_{med_{MAX}}}{2f_{mod}}\right)}{\cos \phi \cdot \cos\left(\frac{\pi f_{med_{MAX}} \phi_{med_{MAX}}}{2f_{mod}}\right) + \sin \phi \cdot \sin\left(\frac{\pi f_{med_{MAX}} \phi_{med_{MAX}}}{2f_{mod}}\right)}$$

$$\tan \phi' = \frac{\tan \phi \cdot \cos\left(\frac{\pi f_{med_{MAX}} \phi_{med_{MAX}}}{2f_{mod}}\right) + \sin\left(\frac{\pi f_{med_{MAX}} \phi_{med_{MAX}}}{2f_{mod}}\right)}{\cos\left(\frac{\pi f_{med_{MAX}} \phi_{med_{MAX}}}{2f_{mod}}\right) + \tan \phi \cdot \sin\left(\frac{\pi f_{med_{MAX}} \phi_{med_{MAX}}}{2f_{mod}}\right)}$$

De esta ecuación se obtiene que la tangente obtenida al procesar el seno y el coseno presenta una desviación de la tangente real que depende tanto del valor de la tangente real como del factor S :

$$S = \frac{f_{mod}}{f_{med_{MAX}} \phi_{med_{MAX}}} \quad (3.15)$$

Este factor puede entenderse como el sobremuestreo utilizado en la medida. Idealmente, debe ser infinito (cuanto mayor sea, menor será la diferencia de error). Relacionando este factor con la frecuencia mínima de muestreo anterior, su valor debe ser: $S > 4$.

En la **Figura 18a** y **Figura 18b** se representa la tangente obtenida en función de la tangente real para diferentes factores, y la diferencia de error entre ambas en la **Figura 18e**. Lo mismo se representa en la **Figura 18c**, **Figura 18d** y **Figura 18f** para la fase óptica.

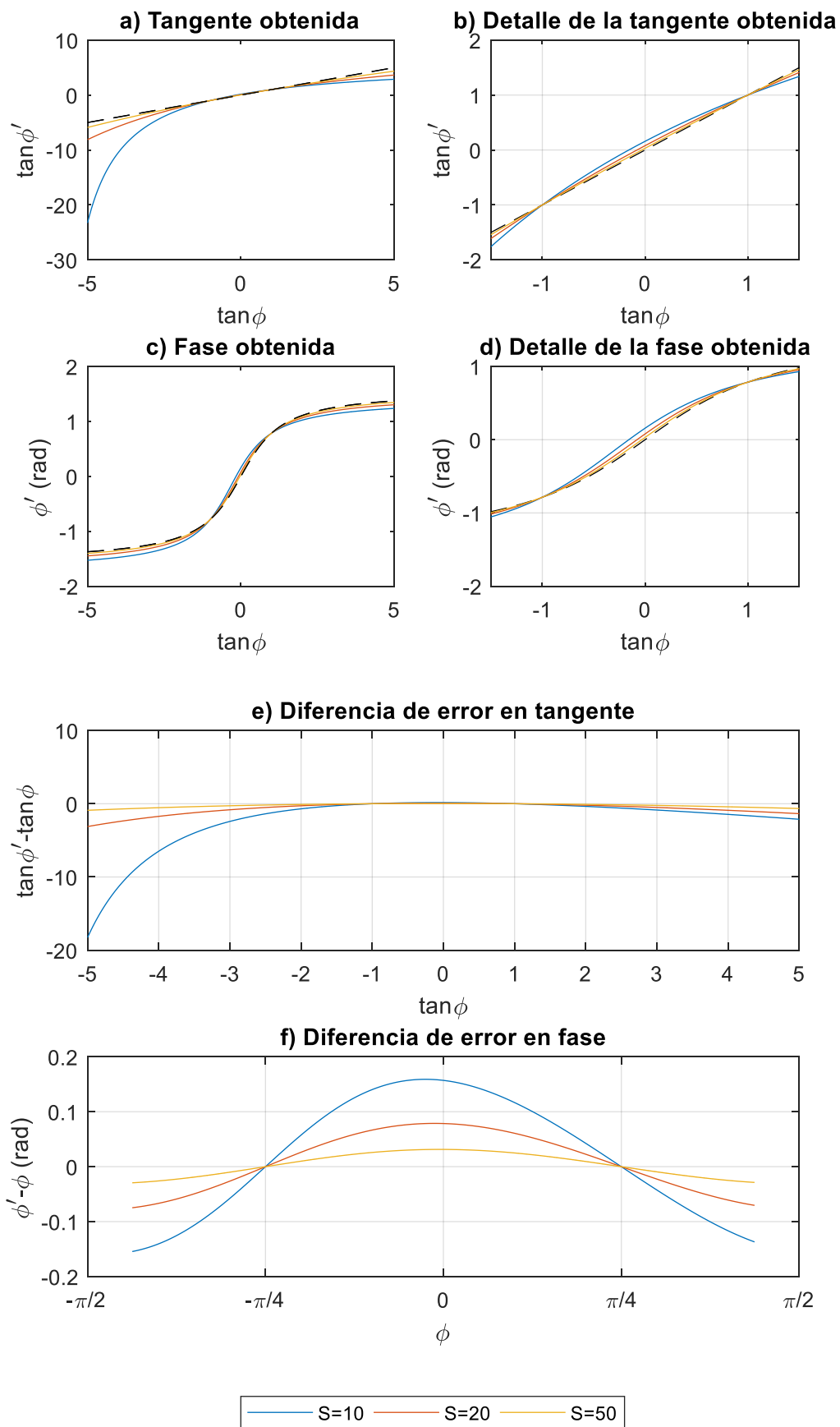


Figura 18. Error en el cálculo de la tangente y la fase

En la **Figura 18** no se representa la asíntota vertical que se produce cuando se anula el denominador. No obstante, aunque esto pudiera suponer un problema, la función sigue siendo continua en fase al reconstruirla ya que el salto se produce de $-\frac{\pi}{2}$ a $\frac{\pi}{2}$ radianes y el error no aumenta. Esta asíntota se desplaza hacia la izquierda al aumentar el factor S , al mismo tiempo que se reduce el error máximo.

Aunque en principio el error obtenido es variable (que en el mayor de los casos no será apreciable y en este trabajo no se tiene en cuenta), en la medición ocurren dos cosas. En primer lugar, normalmente las señales medidas presentan simetría y más concretamente en las vibraciones. De esta manera, el error no se suma en la fase total acumulada (se compensa el error del hemicycle positivo con el negativo). En segundo lugar, existen dos fases ($-\frac{\pi}{4}$ y $\frac{\pi}{4}$ radianes) para las cuales el error es nulo. Conociendo esta información y otros puntos en los que el error sea prácticamente nulo al no existir variación entre dos fases instantáneas consecutivas (pendiente nula), se podría llegar a construir una curva de forma incluso más precisa repitiendo el cálculo de la fase óptica acumulada a partir de la información proporcionada por la diferencia entre las fases ópticas instantáneas.

En resumen, se debe tener en cuenta principalmente la frecuencia mínima de muestreo para reconstruir correctamente la fase óptica total acumulada con $S > 4$. Además, dependiendo de la precisión necesaria este factor debe ser aumentado, consiguiendo un menor ruido a la salida (menos armónicos residuales y más alejados en la FFT final). Un valor razonable en cuanto a precisión y reconstrucción de la fase es $S = 50$, con el que se obtiene un error máximo de 0.03 radianes que equivale aproximadamente a un 1% de error a fondo de escala y que se reduce a un 0.5% si se aplican las reglas de signo del apartado anterior (sin tener en cuenta la limitación de la resolución de los instrumentos de adquisición y procesamiento utilizados).

3.4. SISTEMA COMPLETO PROPUESTO

El diagrama de bloques del sistema completo propuesto se representa en la **Figura 19**.



Figura 19. Diagrama de bloques del sistema completo de interferometría

Una vez comprobado el funcionamiento de la técnica de detección síncrona, es posible diseñar un sistema completo de interferometría que integre la técnica desarrollada. El objetivo del sistema completo será principalmente automatizar las tareas descritas en los apartados anteriores. Dos tareas importantes son la calibración del interferómetro a la profundidad de modulación correcta de la fase óptica y la adquisición síncrona en los máximos y mínimos de la señal de modulación. Realizar la primera tarea manualmente puede resultar complicado en determinadas situaciones según la estabilidad del interferómetro y la influencia de los factores externos, dificultando la distinción de la fase óptica de modulación introducida a la salida. Por ello se ha decidido automatizar el proceso mediante el uso de un microcontrolador. Por otro lado, automatizar la segunda tarea posibilita no almacenar más datos de los necesarios. Adquirir únicamente en los máximos y mínimos de la señal de modulación reduce en gran medida la cantidad de memoria requerida para almacenar los valores de la señal de salida, siendo necesario únicamente adquirir los valores de esta en estos instantes y ninguno de la señal de modulación.

Resumidamente, se parte de un sistema óptico que consta de un interferómetro de Mach-Zehnder que es excitado y cuya respuesta varía en función de las magnitudes físicas que se desean medir. Su respuesta se obtiene mediante un circuito electrónico de acondicionamiento que también se encarga de excitar el sistema. Las señales de este circuito son proporcionadas o adquiridas a través de un microcontrolador que a su vez procesa la información recibida y envía los resultados a un ordenador, donde finalmente pueden ser visualizados y registrados.

En el primer apartado de este capítulo se ha presentado la instalación del sistema óptico utilizado en el laboratorio que corresponde al primer subsistema del sistema completo. Su diagrama de bloques se muestra en la **Figura 20**, de manera que el mismo está conectado con bloques del sistema de acondicionamiento en el que se acondicionan tanto la señal de modulación como la señal de salida para obtener finalmente los resultados deseados.

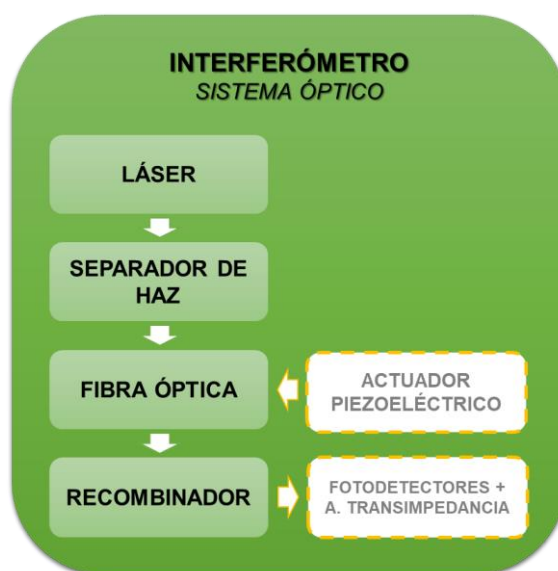


Figura 20. Diagrama de bloques del sistema óptico (interferómetro)

En el siguiente capítulo se detallan los tres últimos bloques del sistema completo: el procedimiento seguido para el diseño, la implementación y la unión de todos ellos.

4. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

4.1. ACONDICIONAMIENTO

El diagrama de bloques de este subsistema y su relación con el resto se muestra en la **Figura 21**.

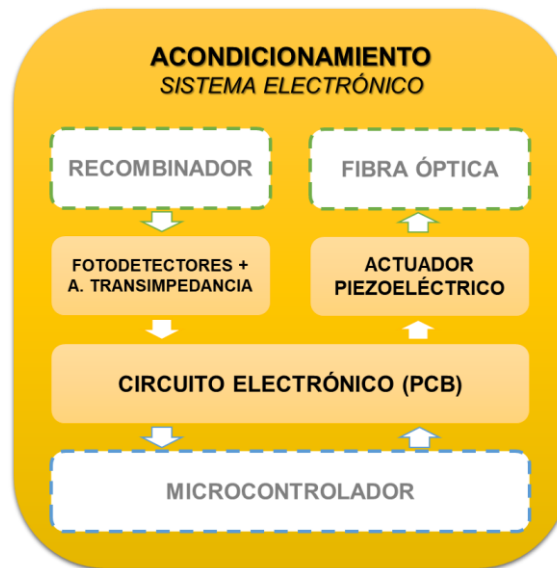


Figura 21. Diagrama de bloques del sistema electrónico (acondicionamiento)

El objetivo del sistema electrónico de acondicionamiento es por un lado convertir la señal óptica de salida del interferómetro en una señal de tensión eléctrica para adquirirla en un microcontrolador y por otro lado acondicionar la señal de modulación aplicada al PZT. En los siguientes apartados se exponen los circuitos electrónicos diseñados y la placa de circuito impreso que los implementa.

4.1.1. CONVERSIÓN DE LA POTENCIA ÓPTICA A SEÑAL ELÉCTRICA

Para convertir la potencia óptica en el extremo final de la fibra óptica en una señal de tensión eléctrica se debe hacer uso de fotodiodos.

Concretamente, en este proyecto se ha utilizado un aparato comercial que consiste en dos fotodiodos con un amplificador de transimpedancia (fotodiodos amplificados balanceados) de ganancia fija que genera una tensión a la salida proporcional a la diferencia entre las dos fotocorrientes generadas en los dos fotodiodos, es decir, proporcional a la diferencia entre la potencia óptica de las dos señales ópticas de entrada (para ello es importante que los fotodiodos

trabajen en su zona lineal). El modelo es THORLABS PDB410A³ (**Figura 22**) y como características importantes presenta un bajo ruido y es capaz de soportar frecuencias de hasta 100 MHz.



Figura 22. THORLABS PDB410A con dos fotodiodos y un amplificador de transimpedancia de ganancia fija (fotodiodos amplificados balanceados) [17]

El recombinador proporciona a la salida la señal óptica de interferencia en dos brazos de fibra óptica, pero con el término de interés con signos opuestos. Por tanto, sería posible utilizar únicamente un fotodiodo. Sin embargo, se aprovechan los dos fotodiodos y de esta manera se obtiene el doble de tensión a la salida al ser la diferencia entre ambas señales ópticas igual al doble de una de ellas, eliminando también la componente continua de ambas señales. Esto es una ventaja, ya que la señal de interferencia puede ser débil en el montaje y por tanto conviene obtener la máxima amplitud posible.

En la **Figura 23** se muestra el diagrama funcional del aparato en el que se indican las dos señales ópticas de entrada ("Input+" y "Input-") y la señal de tensión eléctrica de salida ("RF Output"). Esta última es acondicionada para ser adquirida más adelante.

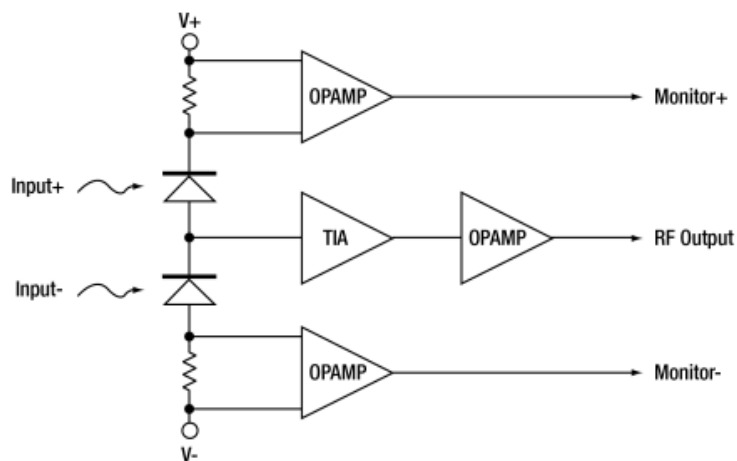


Figura 23. Diagrama funcional de THORLABS PDB410A [17]

³ Hoja de características:

<https://www.thorlabs.com/thorproduct.cfm?partnumber=PDB410A&pn=PDB410A>

4.1.2. ACONDICIONAMIENTO DE LA SEÑAL DE MODULACIÓN DEL PZT

Este circuito y el expuesto en el siguiente apartado formarán parte de la misma placa de circuito impreso, por lo que la alimentación “V+”, “V-” y “0” (GND) es común a ambos y procede de los reguladores de tensión de la **Figura 24**. Las tensiones de salida de los reguladores (positiva y negativa) son de 12 V.

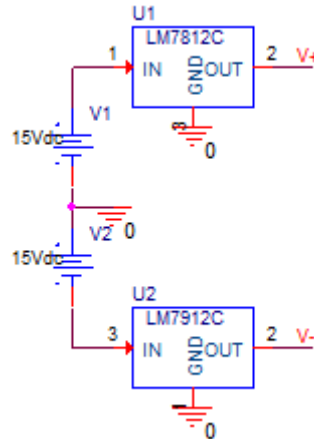


Figura 24. Alimentación de los circuitos (reguladores de tensión)

Para el acondicionamiento de la señal de modulación del PZT se ha diseñado el circuito mostrado en la **Figura 25**. Con este circuito se busca obtener una señal alterna pura de amplitud variable que se aplicará al PZT a partir de la señal de modulación proveniente del microcontrolador (cuya generación se explica más adelante).

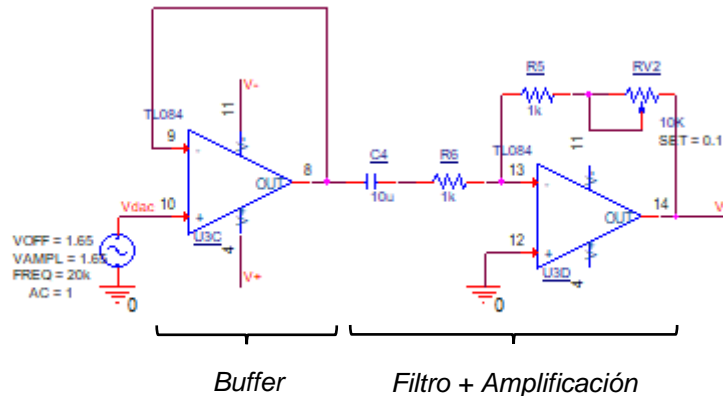


Figura 25. Circuito electrónico de acondicionamiento de la señal de modulación del PZT

En este circuito:

- **Vdac**: señal de modulación proveniente del microcontrolador.
- **Vi**: señal de modulación acondicionada que se aplica al PZT.

El primer bloque de este circuito consiste en una etapa de aislamiento (*buffer*) para evitar problemas con la impedancia de salida del microcontrolador.

Además, la señal proveniente del microcontrolador presenta una componente continua que debe filtrarse. Para ello se hace uso de un filtro activo paso alto en la segunda etapa. En el lazo de realimentación se ha incorporado además un potenciómetro que permita amplificar la señal dado que, aunque en el microcontrolador esta se pueda variar, la amplitud máxima pico a pico de una señal alterna en el pin de salida es de $3.3 V_{pp}$.

En la **Figura 26** se representa la simulación del circuito, es decir, la evolución temporal de las tensiones “Vdac” y “Vi” para una tensión “Vdac” sinusoidal de frecuencia $f_{V_{DAC}} = 20 \text{ kHz}$ con componente continua (la que se obtendría del microcontrolador). El potenciómetro se ha ajustado de manera que la ganancia de la segunda etapa sea $G = 2 \text{ V/V}$, es decir, $RV2 = 1 \text{ k}\Omega$.

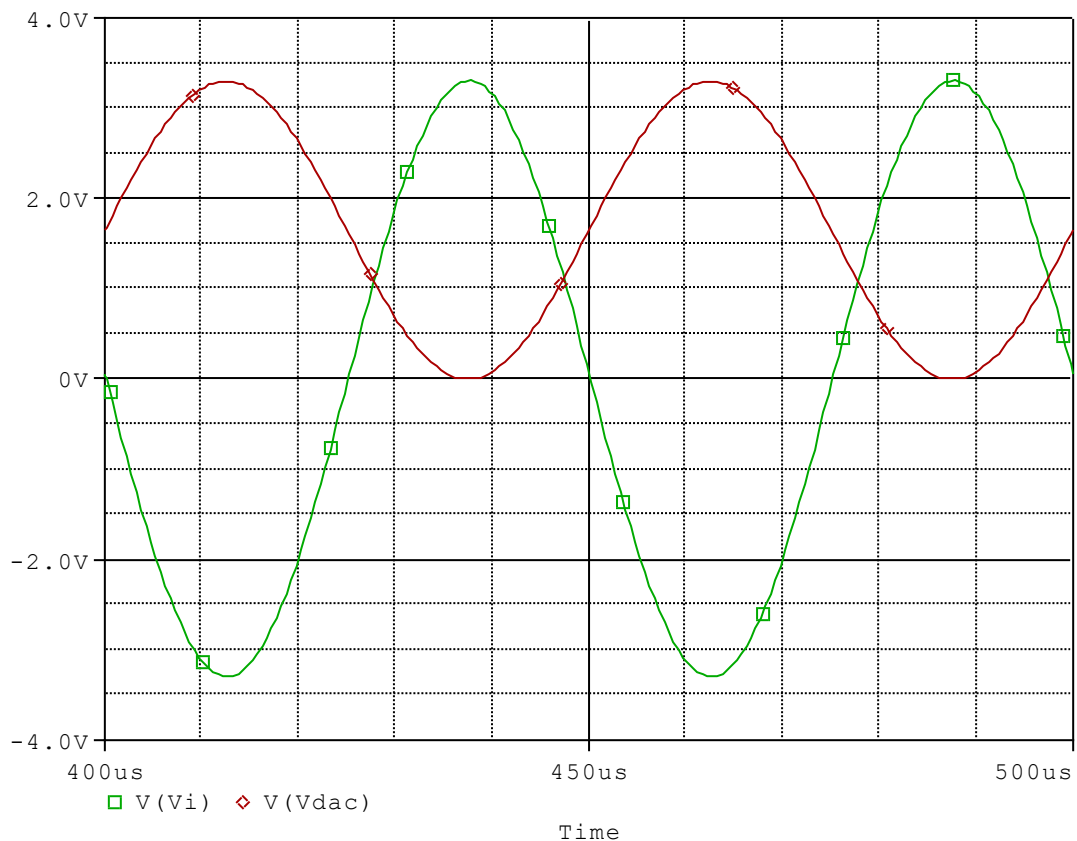


Figura 26. Simulación del circuito de acondicionamiento de la señal de modulación del PZT

Una vez acondicionada la tensión que se va a aplicar al PZT, este actuador también debe ser acondicionado para excitar la fibra óptica. Para ello se ha enrollado la fibra óptica sobre el mismo de manera que las microdeformaciones del material piezoeléctrico al aplicarle tensión eléctrica también deformen la fibra óptica adherida modificando la fase óptica de modulación a la salida. El PZT enrollado se muestra en la **Figura 27**. El PZT debe enrollarse con el máximo número de vueltas posible con el objetivo de amplificar el efecto piezoeléctrico generado. Un mayor número de vueltas implica una menor tensión eléctrica necesaria para obtener la misma diferencia de fase óptica a la salida. La relación entre la variación de la fase óptica a la salida y la tensión aplicada sigue aproximadamente una tendencia lineal.

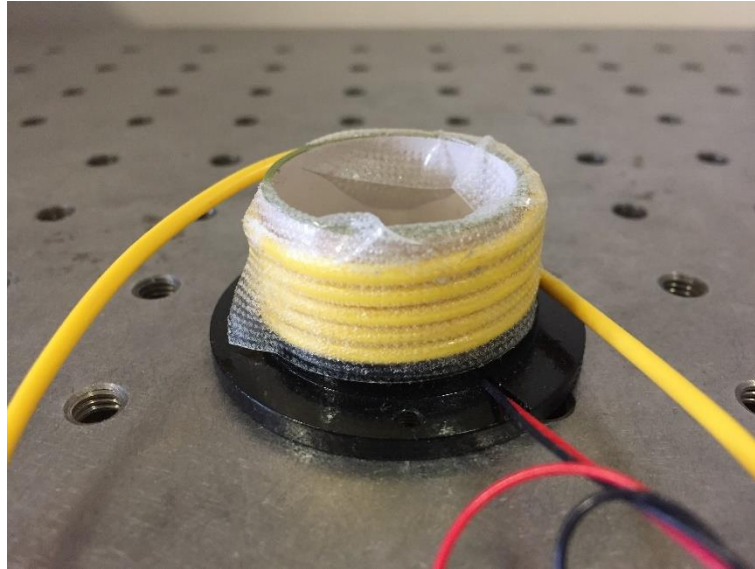


Figura 27. Fibra óptica enrollada sobre el PZT

4.1.3. ACONDICIONAMIENTO DE LA SEÑAL DE SALIDA DEL INTERFERÓMETRO

Para el acondicionamiento de la señal de salida del interferómetro se ha diseñado el circuito mostrado en la **Figura 28**. El propósito de este circuito es obtener una señal alterna con componente continua en el rango de tensión admitido por el microcontrolador para adquirir así la señal de salida del interferómetro proporcionada por el amplificador de transimpedancia del aparato comercial anteriormente descrito. La técnica de adquisición se explica más adelante.

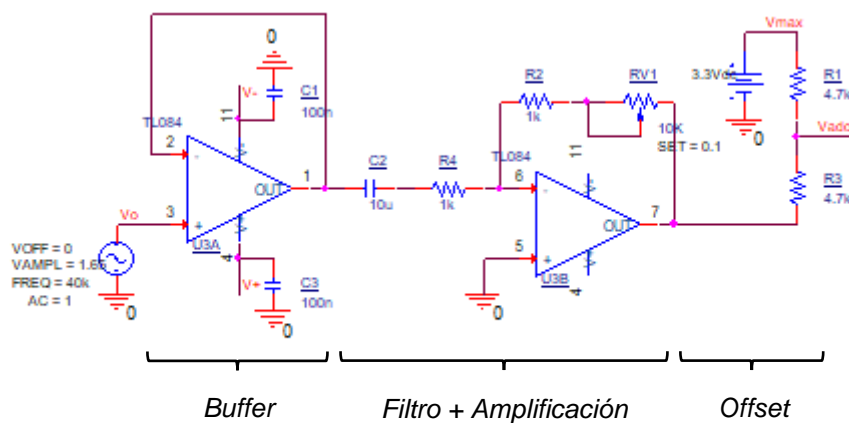


Figura 28. Circuito electrónico de acondicionamiento de la señal de salida del interferómetro

En este circuito:

- **Vo**: señal de salida del amplificador de transimpedancia.
- **Vadc**: señal de salida acondicionada para adquirirla en el microcontrolador.
- **Vmax**: tensión máxima de adquisición del microcontrolador.

La primera etapa tiene la misma finalidad que en el circuito anterior, es decir, aislar el amplificador de transimpedancia del resto del circuito.

La segunda etapa también se trata de un filtro activo paso alto cuyo fin es filtrar la componente continua de la tensión de salida del amplificador de transimpedancia. Normalmente, esta componente es nula al usar los fotodiodos amplificados balanceados. Sin embargo, si la potencia óptica del láser no se divide igualmente en los dos brazos del interferómetro entonces esta se crea y se debe eliminar (el procesamiento final no sería correcto si se mantiene). También se añade un potenciómetro para regular la amplitud de la señal que se adquirirá en el microcontrolador.

Por último, la tercera etapa es una etapa de corrección de offset que añade una componente continua igual a la mitad de la tensión máxima posible de adquirir en el microcontrolador. La tensión máxima admitida por el microcontrolador se indica en el circuito proporcionando esa tensión en “Vmax”. En este caso, la tensión máxima admitida es 3.3 V (rango de 0 a 3.3 V). Cuando la señal de salida del amplificador de transimpedancia es nula, en la salida del circuito se obtiene la mitad de la tensión máxima admitida por el microcontrolador.

En la **Figura 29** se representa la simulación del circuito, es decir, la evolución temporal de las tensiones “Vadc” y “Vo” para una tensión “Vo” sinusoidal con una frecuencia igual al doble de frecuencia de modulación anterior ($f_{Vo} = 40 \text{ kHz}$), que será la frecuencia de adquisición del microcontrolador. El potenciómetro se ha ajustado de manera que la ganancia de la segunda etapa sea $G = 2 \text{ V/V}$, es decir, $RV1 = 1 \text{ k}\Omega$.

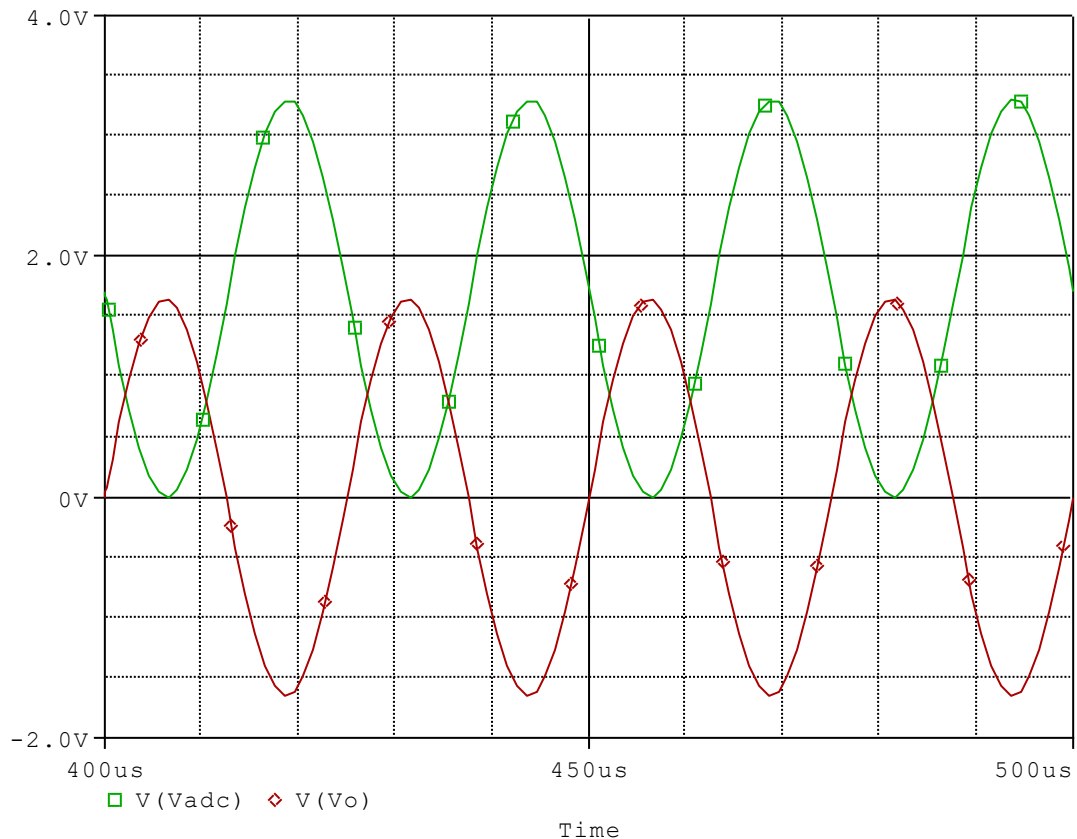


Figura 29. Simulación del circuito de acondicionamiento de la señal de salida del interferómetro

Se observa que la ganancia final es unitaria a pesar de configurar una ganancia $G = 2 \frac{V}{V}$. Esto se debe a que la etapa de corrección de offset actúa como un divisor de tensión, reduciéndola a la mitad.

4.1.4. PLACA DE CIRCUITO IMPRESO

Ambos circuitos se han integrado en una placa de circuito impreso (PCB, *Printed Circuit Board*) que permita una mayor rapidez en las señales, mayor compatibilidad electromagnética y una mayor portabilidad del sistema completo. El *layout* se ha diseñado con el programa KiCad⁴ y se muestra en la **Figura 30**.

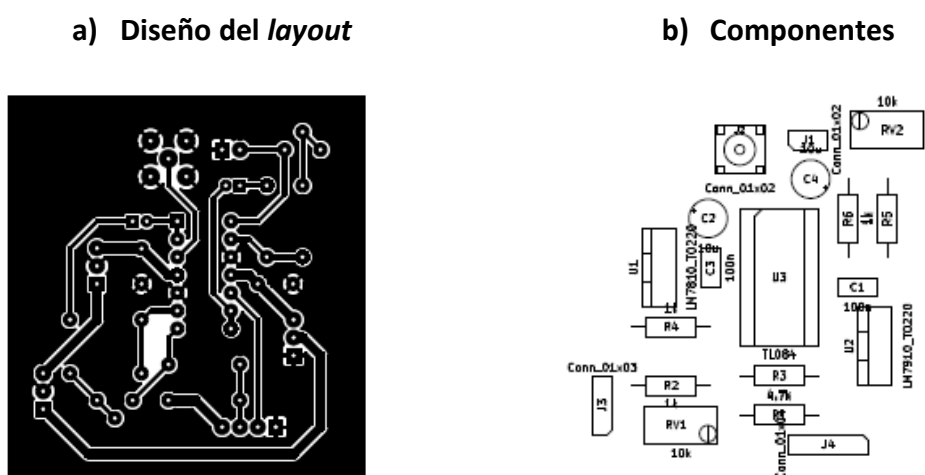


Figura 30. Diseño del layout (a) y componentes (b) de la PCB con dimensiones (ancho x alto): 50.8 x 54.6 mm

Los componentes electrónicos utilizados en la PCB son los presentes en los tres esquemáticos anteriores, y se listan en la **Tabla 2**.

Tabla 2. Componentes electrónicos de la PCB

COMPONENTE	UNIDADES
LM7810 (regulador de tensión positiva)	1
LM7910 (regulador de tensión negativa)	1
TL084 (circuito integrado con 4 amplificadores operacionales)	1
Resistencia de 4.7 k Ω	2
Resistencia de 1 k Ω	4
Potenciómetro de 10 k Ω	2
Condensador de 100 nF	2
Condensador de 10 μ F	2

⁴ Más información: <http://kicad-pcb.org/>

El resultado final se muestra en la **Figura 31**.

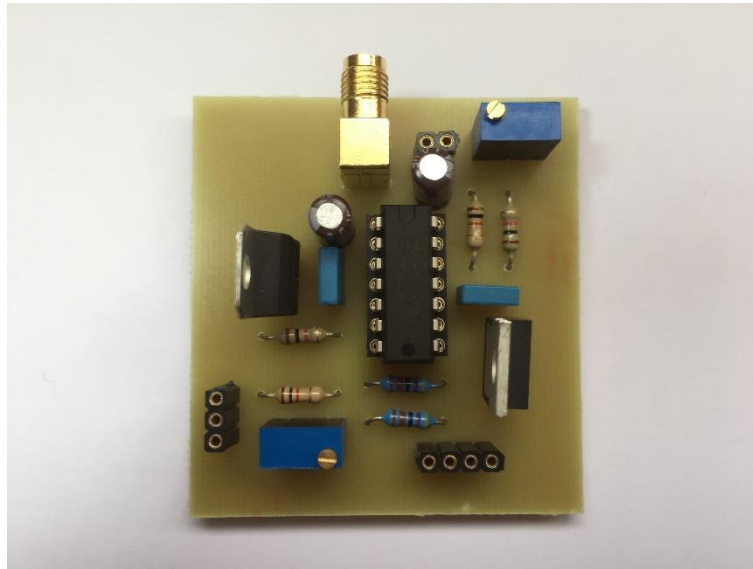


Figura 31. Resultado final de la PCB

4.2. CONTROL Y ADQUISICIÓN

El diagrama de bloques de este subsistema y su relación con el resto se muestra en la **Figura 32**.

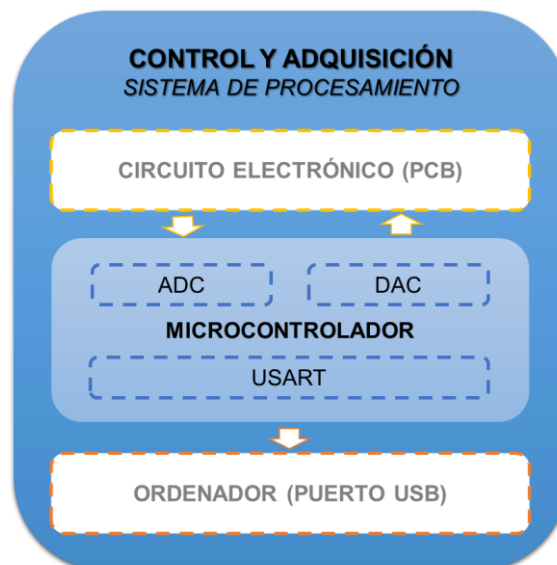


Figura 32. Diagrama de bloques del sistema de control y adquisición (procesamiento)

Se encarga de proporcionar la señal de modulación, calibrar el interferómetro automáticamente y adquirir los valores obtenidos a la salida del interferómetro para procesarlos y obtener finalmente la medida deseada. En los siguientes apartados se expone el diseño del sistema de control y adquisición y las funciones del mismo.

4.2.1. MICROCONTROLADOR

Para realizar el control y la adquisición en el sistema se ha decidido programar un microcontrolador que permita automatizar las tareas necesarias y facilitar así la rápida medición. El microcontrolador debe presentar al menos las siguientes características:

- **Conversor analógico-digital** (ADC, *Analog-to-Digital Converter*) con capacidad de adquirir al menos al doble de la frecuencia de modulación del sistema y con suficiente resolución, considerando una adquisición síncrona en los máximos y mínimos.
- **Conversor digital-analógico** (DAC, *Digital-to-Analog Converter*) con capacidad de emitir una señal sinusoidal con suficiente resolución a la frecuencia de modulación y cuya amplitud pueda ser variable.
- **Capacidad de cálculo relativamente complejo:** transformada de Fourier discreta (DFT, *Discrete Fourier Transform*) y arcotangente.
- **Comunicación con un ordenador** a través de un puerto USB para obtener los resultados de la medición y configurar los parámetros del sistema.
- **Memoria** suficiente para almacenar un número razonable de muestras y procesarlas.

Estos requisitos se consideran cumplidos con el microcontrolador STM32L152RE⁵ de *ST Microelectronics*. Se trata de un microcontrolador con un microprocesador *ARM Cortex M3* de 32 bits, encapsulado de 64 pines y los periféricos necesarios para realizar las tareas necesarias en el sistema con un bajo consumo de energía.

En este proyecto se ha utilizado la placa de desarrollo NUCLEO-L152RE del mismo fabricante mostrada en la **Figura 33** que incluye el microcontrolador mencionado, debido a que incorpora conectores para los pines del microcontrolador y un puerto USB que posibilita la comunicación con el ordenador y su programación.

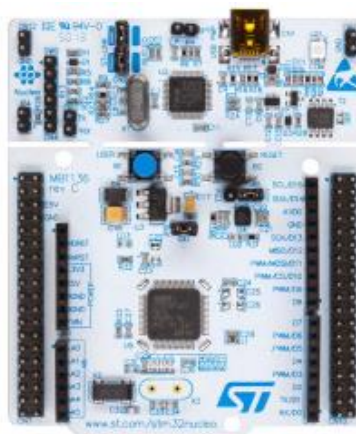


Figura 33. Placa de desarrollo NUCLEO-L152RE con el microcontrolador STM32L152RE [18]

⁵ Hoja de características:

<http://www.st.com/content/ccc/resource/technical/document/datasheet/group1/a7/13/6a/ce/1f/f3/40/c1/DM00098321/files/DM00098321.pdf/jcr:content/translations/en.DM00098321.pdf>

El microcontrolador presenta una gran variedad de periféricos integrados, de los cuales se han aprovechado los siguientes para realizar las funciones necesarias:

- **ADC:** encargado de adquirir la señal de salida del interferómetro del circuito electrónico de acondicionamiento explicado anteriormente.
- **DAC:** encargado de emitir la señal de modulación o excitación del PZT, que posteriormente es acondicionada en el circuito electrónico de acondicionamiento para eliminar la componente continua.
- **DMA** (*Direct Memory Access*): encargado de transmitir la información de la señal sinusoidal de modulación contenida en la memoria (valores instantáneos en datos de 12 bits) al DAC mediante acceso directo a la memoria.
- **TIM2:** este temporizador o *timer* actúa como disparo o *trigger* del DMA para cambiar de dato y construir la señal de modulación a la salida del DAC.
- **TIM9:** este *timer* se encadena al TIM2 posibilitando la adquisición síncrona en los máximos y mínimos de la señal de modulación.
- **USART2** (*Universal Synchronous and Asynchronous Receiver-Transmitter*): posibilita la comunicación serie con el ordenador (u otros dispositivos) a través de un cable USB (o dos conexiones de recepción y transmisión).

4.2.2. GENERACIÓN DE LA SEÑAL DE MODULACIÓN

La señal que se debe generar es sinusoidal y de amplitud variable para posibilitar la calibración automática del sistema. Por diseño se han escogido 50 puntos uniformemente distribuidos en el periodo de la señal para diferentes amplitudes que se almacenan en una tabla en la memoria del microcontrolador (técnica “*look-up table*”), de manera que estos datos sean transmitidos al DAC y se genere la señal en el pin de salida del mismo.

Puesto que es deseable que el microprocesador no se encuentre ocupado todo el tiempo transmitiendo datos de la memoria al DAC, se ha aprovechado el DMA que posibilita liberar al microprocesador de dicha tarea. Este periférico se encarga de transmitir continuamente en segundo plano los datos de la tabla al DAC mediante acceso directo a la memoria para construir la señal sinusoidal buscada.

Por un lado, se debe configurar la frecuencia de la señal. El evento de actualización del TIM2 actúa como *trigger* del DMA para transmitir un nuevo dato al DAC generando así la señal de modulación. La frecuencia de este evento de actualización f_{UEV} viene dada por la siguiente expresión, sabiendo que la frecuencia de reloj del *bus* al que está conectado TIM2 es $f_{CLK} = 32$ MHz:

$$f_{UEV} = \frac{f_{CLK}}{(P + 1)(A + 1)} \quad (4.1)$$

donde P es el preescalado y A es la autorecarga.

Por tanto, teniendo en cuenta que existen 50 datos almacenados del periodo de la señal de modulación, la frecuencia de la señal de modulación f_{DAC} y escogiendo un preescalado nulo ($P = 0$) se obtiene:

$$f_{DAC} = \frac{f_{UEV}}{50} = \frac{f_{CLK}}{50(P+1)(A+1)} \xrightarrow{P=0} f_{DAC} = \frac{f_{CLK}}{50(A+1)} \rightarrow A = \frac{f_{CLK}}{50f_{DAC}} - 1$$

$$A = \frac{640000}{f_{DAC}} - 1 \quad A \in [1,65535]$$
(4.2)

De esta manera se obtiene el valor de la autorrecarga A en función de la frecuencia de modulación f_{DAC} deseada, teniendo en cuenta los límites (autorrecarga de 16 bits).

Por otro lado, el rango de amplitudes máximo de la señal sinusoidal en el pin de salida es, por diseño, de 100 mV_{pp} a 3.1 V_{pp} respetando un margen de 100 mV en los valores límite de tensión en el pin de salida del DAC (de 0 a 3.3 V). Esta señal es amplificada después y su componente continua también es filtrada en el circuito electrónico de acondicionamiento antes de aplicarla al PZT. Por tanto, el número de amplitudes posibles a la salida es 31, existiendo una diferencia de amplitud pico a pico entre ellas de 100 mV .

De esta manera, para calcular el nivel del DAC (N_{ij}) correspondiente a la señal se aplica la siguiente ecuación a los 50 datos a almacenar para las 31 amplitudes disponibles obteniendo así la tabla de dimensiones 31 x 50, teniendo en cuenta que el DAC presenta una resolución de 12 bits (4096 niveles):

$$N_{ij} = \frac{i}{31} \cdot \cos\left(\frac{j}{50} \cdot 2\pi\right) \cdot 1924 + 2048$$
(4.3)

donde i es el número de amplitud (de 1 a 31) y j es el número de dato (de 1 a 50).

La **Figura 34a** representa los 50 puntos del periodo de la señal sinusoidal para la amplitud máxima posible indicando los instantes en los que el ADC muestreará la señal de salida del interferómetro (puntos rojo y azul) coincidentes con el máximo y mínimo exactos de la señal de modulación, mientras que en la **Figura 34b** se representa la señal generada para las 31 amplitudes posibles, observándose una pequeña diferencia entre amplitudes que permitirá una mayor precisión en la calibración automática.

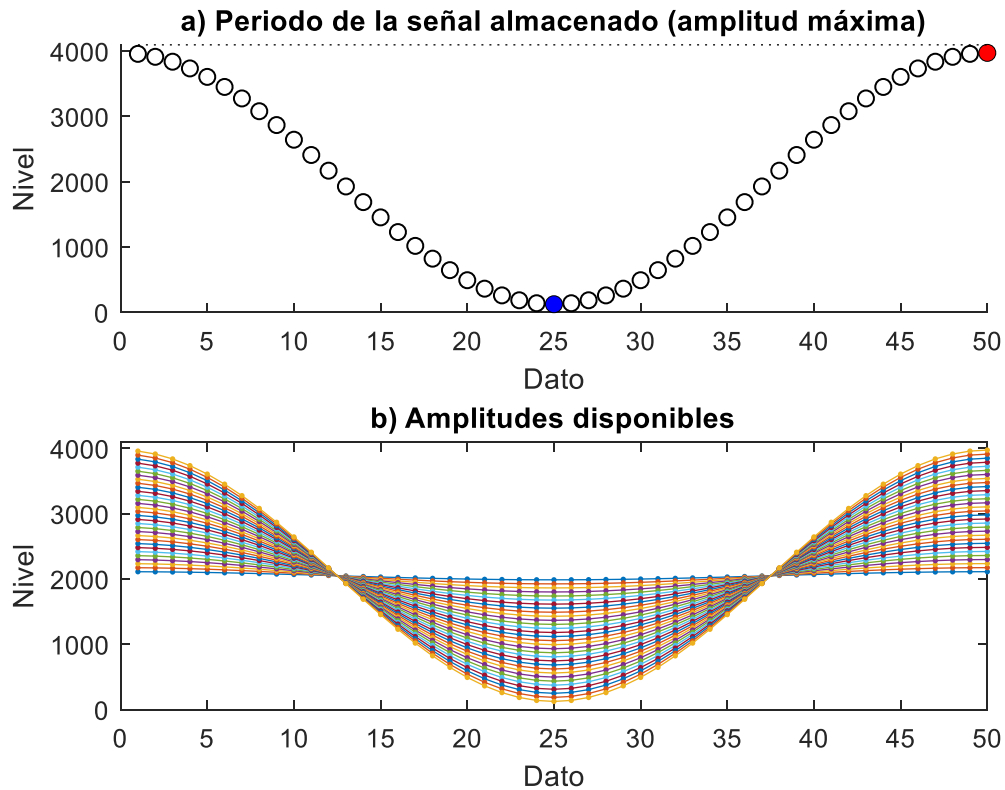


Figura 34. Puntos del periodo de la señal de modulación almacenados (a) y amplitudes disponibles a la salida (b)

4.2.3. ADQUISICIÓN DE LA SEÑAL DE SALIDA

La adquisición de la señal de salida del interferómetro (después de ser acondicionada) se realiza de forma síncrona a la señal de modulación enviada por el mismo microcontrolador.

Para tal propósito se han encadenado los dos temporizadores mencionados anteriormente (TIM2 y TIM9) de manera que su sincronización permita la adquisición exactamente en los instantes de interés de la señal de modulación (máximos y mínimos).

El TIM9 se encadena al TIM2 (utilizado para generar la señal de modulación). El TIM2 se encuentra configurado en modo de cuenta ascendente (*upcounting mode*). Sin embargo, el TIM9 se configura en modo de cuenta centrado (*center-aligned mode*) permitiendo así distinguir entre un máximo y un mínimo generando un evento tanto en el límite superior (valor de la autorrecarga) como en el inferior (cero) de la cuenta. El encadenamiento da lugar a que f_{UEV} del TIM2 sea la frecuencia de reloj del TIM9, y por tanto aumente su cuenta cada vez que se transmite un nuevo dato al DAC. Teniendo en cuenta que el periodo de la señal consta de 50 datos y que existe un mínimo y un máximo en los datos número 25 y 50 respectivamente (**Figura 34a**), la autorrecarga del TIM9 se configura en 25 ($A = 25$) dando lugar a un evento tanto en la cuenta ascendente (mínimo de la señal de modulación) como en la descendente (máximo de la señal de modulación). Estos eventos actúan como *trigger* del ADC y conociendo el sentido de la cuenta se distinguirá de forma inequívoca cuando se adquiere un máximo o un mínimo permitiendo procesar los datos correctamente.

En la **Figura 35** se muestra el funcionamiento de ambos *timers* encadenados para una señal de modulación de 20 kHz de frecuencia.

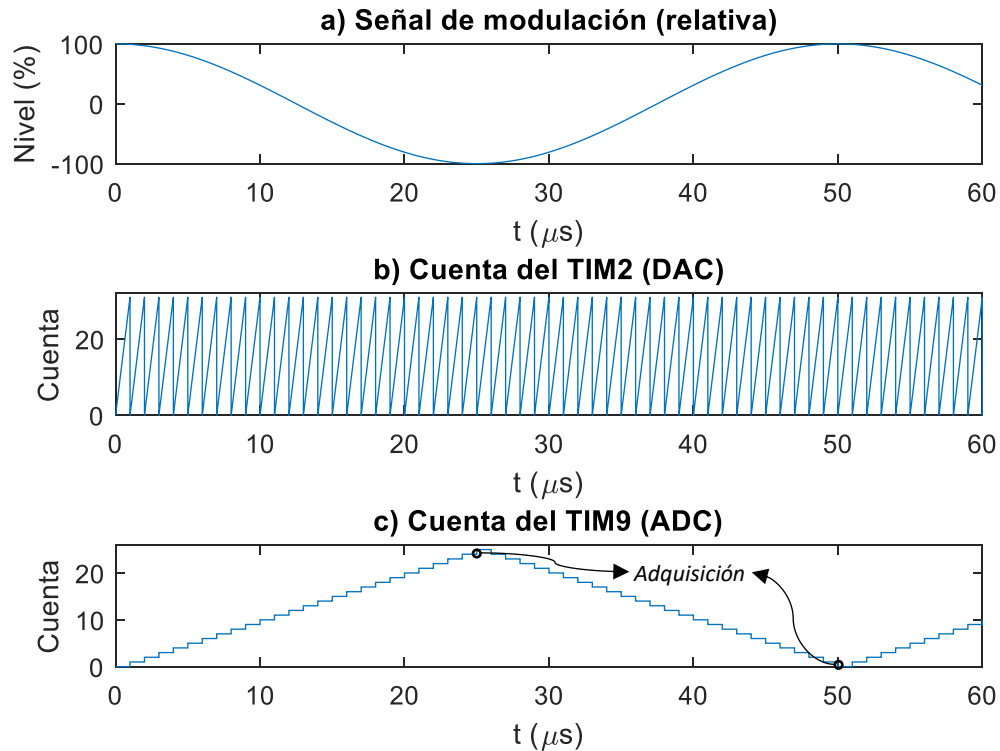


Figura 35. Sincronización entre ambos timers para adquirir en los máximos y mínimos de la señal de modulación

En resumen, cuando el TIM2 llega al final de la cuenta, este incrementa la cuenta del TIM9. Una vez alcanzado el límite superior, la cuenta cambia de sentido y se vuelve decreciente hasta llegar a cero donde se vuelve a cambiar el sentido repitiéndose el proceso de forma cíclica. De esta manera, en los límites superior e inferior se adquiere en el ADC de forma síncrona con la señal de modulación (en los mismos instantes que ocurren sus máximos y sus mínimos).

Por último, también se debe tener en cuenta que los datos adquiridos mediante el ADC se encuentran en un rango de tensión positivo de 0 a 3.3 V. Sin embargo, estos datos se corresponden a la señal de salida del interferómetro acondicionada cuya componente continua es nula originalmente. Por tanto, es necesario eliminar la componente continua para procesar los datos restando 1.65 V, o lo que es lo mismo, restando la mitad de la resolución máxima del ADC, es decir, 2048 niveles.

4.2.4. CALIBRACIÓN AUTOMÁTICA

La calibración manual del sistema depende totalmente de la observación directa de las señales en el osciloscopio evaluando la correcta calibración del sistema en función de la variación de la señal de salida al cambiar la amplitud de la señal de modulación aplicada al PZT. Sin embargo, la calibración manual del sistema presenta una serie de inconvenientes como la baja precisión limitada por la percepción del observador de las señales, la necesidad de un

osciloscopio y el tiempo que lleva realizar esta tarea. Por ello se ha decidido posibilitar la calibración automática del sistema aprovechando las capacidades del microcontrolador.

Para calibrar el sistema se debe conocer la respuesta del mismo al introducir la señal de modulación con una amplitud determinada, la cual se debe modificar para calibrarlo. Para este fin se adquirirá la señal de salida mediante el ADC a la frecuencia correspondiente a obtener 25 muestras en un semiperiodo de la señal (cambiando como *trigger* el TIM9 por el TIM2 momentáneamente).

Al introducir una amplitud de modulación pico a pico de $2\pi + \frac{\pi}{2}$ radianes de la fase óptica (que cumple la ecuación (3.8) que debe satisfacerse para obtener dos señales en cuadratura) se deduce que entre el máximo y el mínimo de la señal de modulación debe producirse la respuesta para el periodo completo de la función coseno más el tiempo correspondiente a una variación de fase de $\frac{\pi}{2}$ radianes (debido a la función de transferencia del interferómetro). En la **Figura 36** se representa la intensidad de salida relativa del interferómetro cuando este se encuentra calibrado en cuadratura para diferentes fases ópticas iniciales ϕ_0 en el intervalo de tiempo entre un máximo y un mínimo consecutivos de la señal de modulación, con una frecuencia de modulación de 20 kHz.

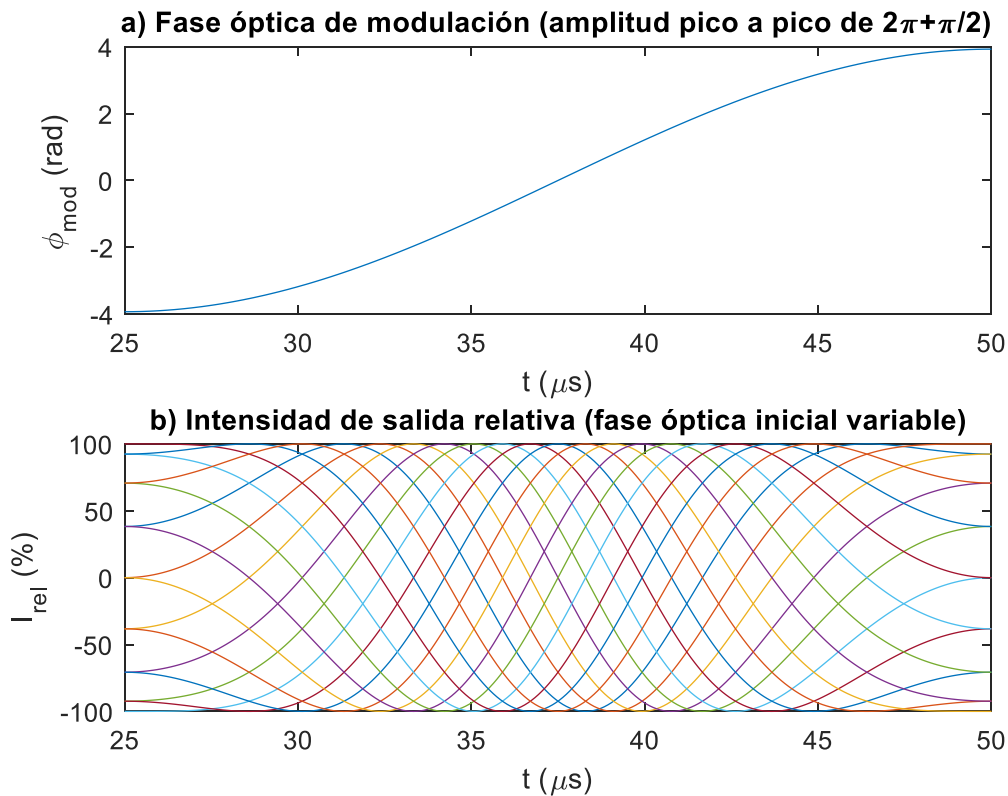


Figura 36. Intensidad de salida relativa del interferómetro calibrado en cuadratura para diferentes fases ópticas iniciales

Por otro lado, en la **Figura 37** se representa la respuesta del sistema para diferentes amplitudes de modulación de la fase óptica y una fase óptica inicial constante: $\phi_0 = 0$, siendo

la intensidad de salida resaltada en negro la correspondiente a la amplitud de modulación correcta (en cuadratura):

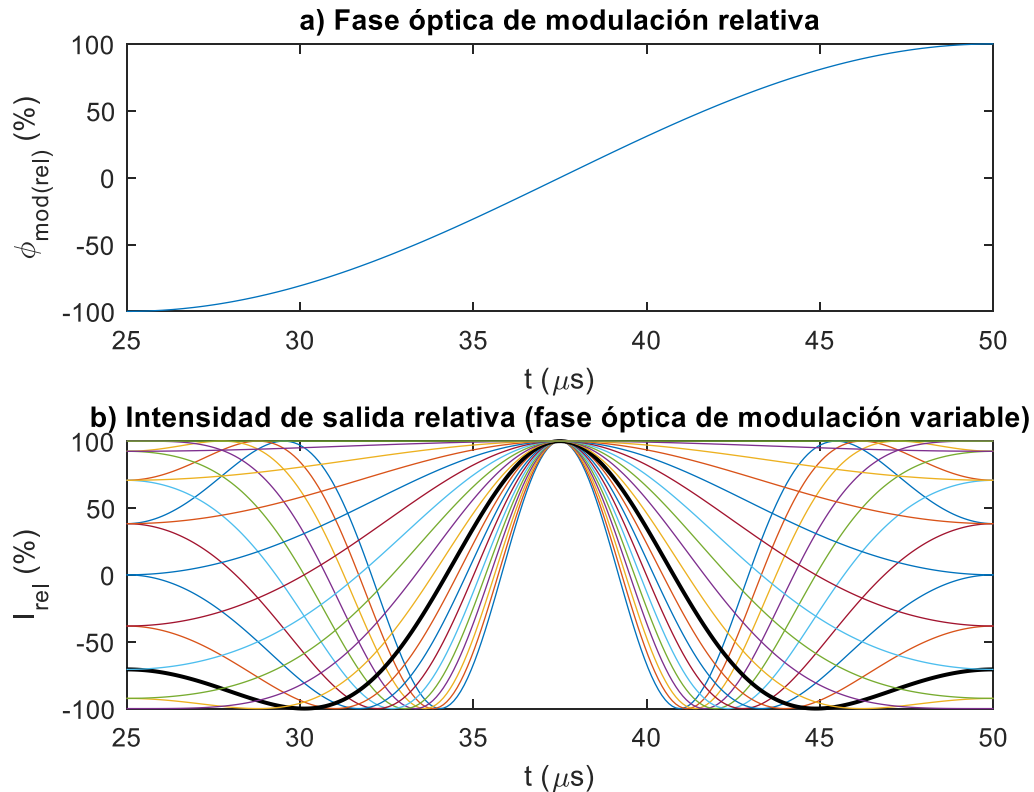


Figura 37. Intensidad de salida relativa del interferómetro para diferentes amplitudes de la fase óptica de modulación

De ambas figuras se obtienen dos conclusiones:

- Por un lado, en la **Figura 36** se puede observar la existencia de un tramo prácticamente lineal en la fase óptica de modulación que provoca una respuesta en el interferómetro conocida independiente de la fase óptica inicial.
- Por otro lado, de la **Figura 37** se obtiene que, a igualdad de fase óptica inicial, la frecuencia de la nueva señal sinusoidal generada como respuesta en el tramo prácticamente lineal varía en función de la amplitud de modulación.

Conociendo estos datos se puede establecer una relación que permita conocer en qué estado de modulación se encuentra el sistema independientemente de su fase inicial y de la no linealidad de la función de transferencia del interferómetro.

Una opción sería calcular la pendiente de la zona lineal de la señal de salida obtenida, característica única para cada una de las amplitudes de modulación (mayor amplitud de modulación implica mayor pendiente). Sin embargo, para ello es necesario conocer el máximo y el mínimo de la señal para normalizar la pendiente y esto no es posible siempre debido a la limitación de la frecuencia de muestreo del ADC. Si se normaliza con un intervalo de intensidad entre máximo y mínimo distinto al real (variación menor), la pendiente obtenida será mayor y

por tanto no se evalúa la calibración de forma precisa. Otra opción sería adquirir varios puntos de la señal de salida y compararla con la función seno o coseno. Sin embargo, el desfase inicial supone un obstáculo importante a la hora de comparar ambas señales.

Por ello se ha decidido utilizar otra técnica. Esta consiste en evaluar los instantes de tiempo en los que la señal de salida cruza por cero, no siendo necesario normalizar ningún valor. El intervalo de tiempo comprendido entre ambos instantes es único para cada amplitud de la fase óptica portadora variable y es suficiente para determinar la calibración del sistema. No obstante, se realiza una aproximación. Puesto que es muy complicado obtener exactamente los instantes en los que la señal cruza por cero (debido también a la limitación de la frecuencia de muestreo), se interpolan linealmente las dos adquisiciones más próximas a este cruce (por encima y por debajo de cero) y se obtiene el valor linealizado en el que la señal de salida vale cero. Cabe destacar que, aunque se realice esta aproximación, los tiempos medidos son suficientemente precisos ya que la linealización no da lugar a un error apreciable puesto que la función seno es aproximadamente una recta en el tramo cercano a cero según la serie de Taylor para el punto $x_0 = 0$:

$$\sin x \approx \sum_{k=0}^n \frac{(-1)^k}{(2k+1)!} x^{2k+1} \xrightarrow{n=0} \sin x \approx x \quad (4.4)$$

El cruce por cero ocurre cuando el ADC detecta la mitad de la tensión de adquisición máxima disponible, ya que la señal de salida del interferómetro es filtrada para eliminar su componente continua y después es acondicionada para adecuarse a la entrada del ADC añadiendo un offset igual a la mitad de esta tensión ($N_{V_{ADC}=0}$):

$$N_{V_{ADC}=0} = \frac{2^{12}}{2} = \frac{4096}{2} = 2048 \quad (4.5)$$

El valor linealizado del instante de tiempo $t_{V_{ADC}=0}$ en el que la señal cruza por cero se obtiene por tanto a partir de la siguiente ecuación:

$$t_{V_{ADC}=0} = t_{V_{ADC}<0} + \frac{t_{V_{ADC}>0} - t_{V_{ADC}<0}}{N_{V_{ADC}>0} - N_{V_{ADC}<0}} (2048 - N_{V_{ADC}<0}) \quad (4.6)$$

Para calcular los instantes de tiempo en los que la señal de salida es nula se iguala a cero la componente alterna I_{AC} de la función de transferencia del interferómetro con el resto de valores normalizados y $\phi_0 = 0$:

$$\begin{aligned} I_{AC} &= I_0(V \cos(\Delta\phi)) = 0 \rightarrow 1 \cdot (1 \cdot \cos(\phi_{mod_0} \cdot \sin(2\pi f_{mod} t))) = 0 \\ \phi_{mod_0} \cdot \sin(2\pi f_{mod} t) &= \frac{\pi}{2} + \pi k \rightarrow \sin(2\pi f_{mod} t) = \frac{\pi}{\phi_{mod_0}} \left(\frac{1}{2} + k \right) \\ t &= \frac{\arcsen\left(\frac{\pi}{\phi_{mod_0}} \left(\frac{1}{2} + k \right)\right)}{2\pi f_{mod}} \end{aligned} \quad (4.7)$$

Los límites de k vienen dados por las siguientes expresiones:

$$\begin{aligned}
-1 &\leq \frac{\pi}{\phi_{mod0}} \left(\frac{1}{2} + k \right) \leq 1 \\
-\left(\frac{\phi_{mod0}}{\pi} + \frac{1}{2} \right) &\leq k \leq \frac{\phi_{mod0}}{\pi} - \frac{1}{2} \\
\left\{ \begin{array}{l} k_{max} = \left\lfloor \frac{\phi_{mod0}}{\pi} - \frac{1}{2} \right\rfloor = \max \left\{ k \in \mathbb{Z} \mid k \leq \frac{\phi_{mod0}}{\pi} - \frac{1}{2} \right\} \\ k_{min} = \left\lceil -\left(\frac{\phi_{mod0}}{\pi} + \frac{1}{2} \right) \right\rceil = \min \left\{ k \in \mathbb{Z} \mid k \geq -\left(\frac{\phi_{mod0}}{\pi} + \frac{1}{2} \right) \right\} \end{array} \right. & \quad (4.8) \\
k &\in [k_{min}, k_{max}]
\end{aligned}$$

Particularizando para una frecuencia de modulación de 20 kHz se obtiene el intervalo de tiempo entre dos cruces por cero consecutivos, que coinciden con los cruces por cero de la curva resaltada en negro en la **Figura 37b**:

$$\left. \begin{array}{l} \phi_{mod0} = \pi + \frac{\pi}{4} \\ f_{mod} = 20000 \text{ Hz} \end{array} \right\} \rightarrow \left\{ \begin{array}{l} k_{max} = \left\lfloor \frac{\pi + \frac{\pi}{4}}{\pi} - \frac{1}{2} \right\rfloor = 0 \\ k_{min} = \left\lceil -\left(\frac{\pi + \frac{\pi}{4}}{\pi} + \frac{1}{2} \right) \right\rceil = -1 \end{array} \right. \quad (4.9)$$

$$\Delta t = t_{k=0} - t_{k=-1} = \frac{\arcsen\left(\frac{\pi}{2\phi_{mod0}}\right) - \arcsen\left(-\frac{\pi}{2\phi_{mod0}}\right)}{2\pi f_{mod}} = 6.5 \mu s$$

Puesto que en el microcontrolador se puede variar la frecuencia de modulación, se realizará el cálculo en el mismo teniendo en cuenta los términos constantes, dependiendo el intervalo de tiempo únicamente de la frecuencia de modulación deseada.

$$\Delta t_{\phi_{mod0}=\pi+\frac{\pi}{4}} = \frac{\arcsen\left(\frac{\pi}{2\phi_{mod0}}\right) - \arcsen\left(-\frac{\pi}{2\phi_{mod0}}\right)}{2\pi f_{mod}} = \frac{0.1309899}{f_{mod}} \quad (4.10)$$

Sabiendo que la calibración automática se realiza desde la amplitud mínima hasta la amplitud máxima se deduce a partir de las ecuaciones anteriores que, hasta la amplitud de modulación deseada, existen únicamente tres cruces por cero como máximo, necesitando al menos dos consecutivos para el cálculo. En la **Figura 38** se muestran los dos casos extremos a la hora de calibrar que dan lugar a un intervalo de tiempo más corto (cuando el intervalo de tiempo medido coincide prácticamente con la zona lineal de la fase óptica de modulación) o más largo (cuando el intervalo de tiempo medido se aleja de la zona lineal de la fase óptica de modulación), por lo que existe un margen de error. En caso de obtener dos intervalos de tiempo al existir tres cruces por cero, se considerará siempre el de menor duración puesto que se corresponderá con el intervalo de tiempo más cercano a la zona lineal de la fase óptica de modulación y por tanto al intervalo de tiempo calculado. El error cometido en la calibración (diferencia entre el intervalo de tiempo mínimo o ideal y el intervalo de tiempo obtenido real) se informará en la aplicación

informática desarrollada para controlar y utilizar el sistema, pudiendo repetir la calibración hasta obtener un margen de error aceptable.

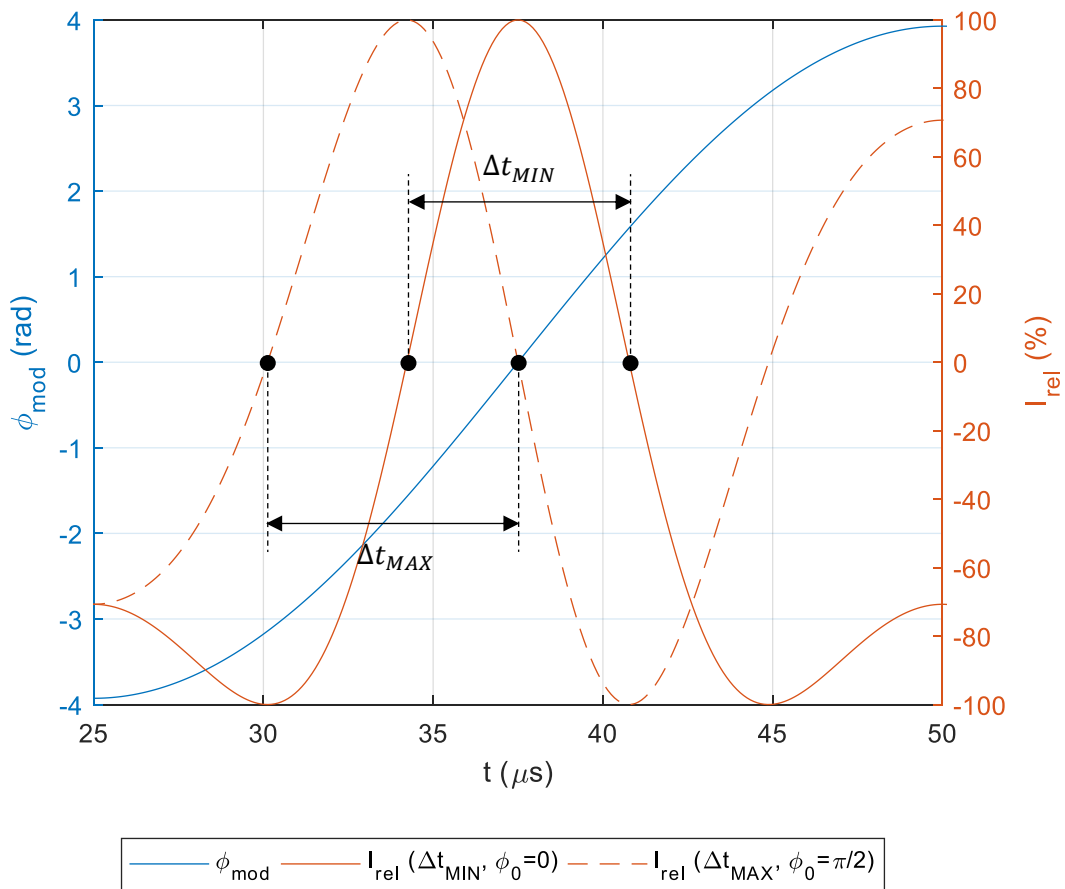


Figura 38. Intervalos de tiempo máximo y mínimo posibles en la calibración en cuadratura

Además, entre las señales puede existir un pequeño retraso, por lo que al implementar el algoritmo de calibración automática en el código se dará por válida la primera amplitud de modulación con la que se obtenga un intervalo de tiempo entre los cruces por cero igual o menor que el intervalo de tiempo calculado.

4.2.5. PROGRAMA, COMUNICACIÓN Y CONEXIONES

El programa desarrollado y cargado en el microcontrolador implementa las ecuaciones y algoritmos explicados en los apartados y capítulos anteriores para controlar el sistema y procesar su respuesta. El código fuente escrito en C se encuentra en el **ANEXO II: Código fuente del microcontrolador**. El entorno de desarrollo integrado empleado es Atollic TrueSTUDIO for STM32⁶ y se ha utilizado el manual de referencia del dispositivo [19].

La ejecución del programa principal en el microcontrolador se muestra en el diagrama de flujo de la **Figura 39**.

⁶ Más información: <https://atollic.com/truestudio/>

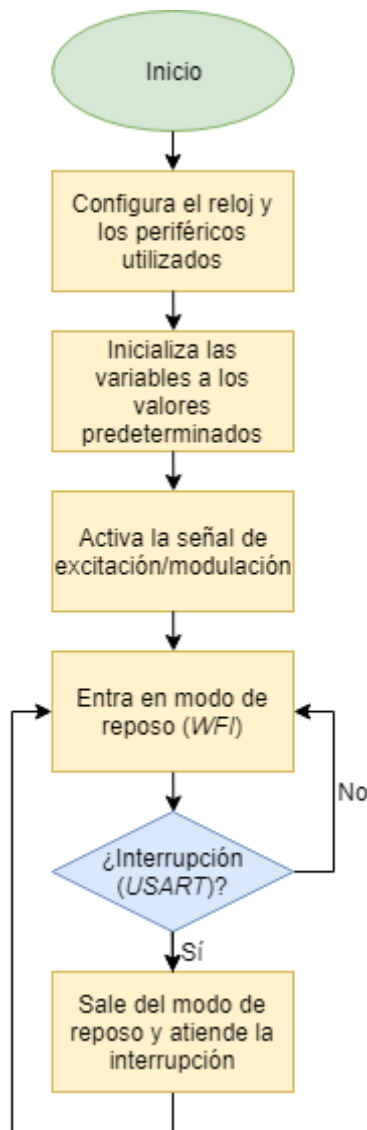


Figura 39. Diagrama de flujo del programa principal del microcontrolador

Una vez alimentado el microcontrolador, este configura los periféricos necesarios anteriormente mencionados e inicializa las variables necesarias, reservando también memoria para las adquirir muestras y procesarlas.

Puesto que es deseable un bajo consumo de energía por parte del dispositivo, este se ha programado para entrar en modo de reposo mientras que no se solicite realizar ninguna medición. No obstante, en modo de reposo el DMA seguirá funcionando en segundo plano emitiendo la señal de modulación en el pin de salida del DAC.

El control del microcontrolador se realiza mediante comunicación serie (USART) a través de un cable USB conectado al ordenador. Para ello se ha diseñado un protocolo de comunicación basado en comandos recogido en el **ANEXO III: Protocolo de comunicación mediante USART**. El ordenador envía un comando que consta de un dato de 8 bits (un carácter) junto a contenido (si es necesario), se evalúa en el microcontrolador, se realizan las acciones correspondientes y se devuelve una respuesta en función del resultado (que puede ser texto o medidas, por ejemplo).

Modificar la frecuencia de modulación, cambiar el número de muestras a adquirir y realizar una medición son ejemplos de comandos que pueden ser enviados al microcontrolador.

Cada vez que se recibe un dato mediante USART se produce una interrupción de manera que el microcontrolador sale del modo de reposo y la atiende ejecutando la Rutina de Atención a la Interrupción (RAI) mostrada en la **Figura 40**. Una vez atendida vuelve al modo de reposo.

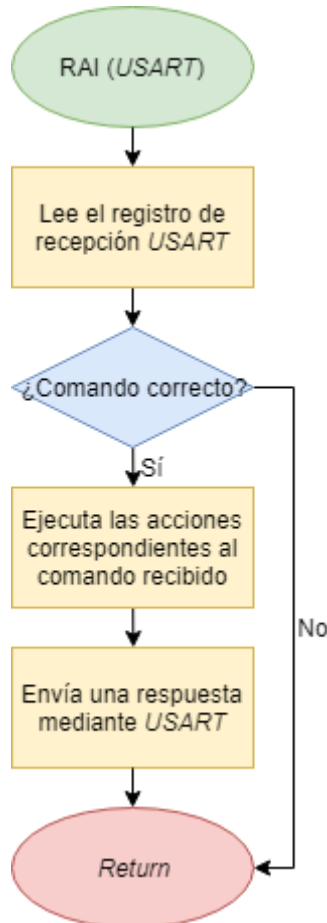


Figura 40. Diagrama de flujo de la RAI de la interrupción por recepción asíncrona USART

Cuando se envía el comando para realizar una medición, el microcontrolador realiza las operaciones indicadas en la **Figura 41** que procesan los valores de la señal de salida del interferómetro adquiridos. Para obtener los resultados se debe enviar otro comando.

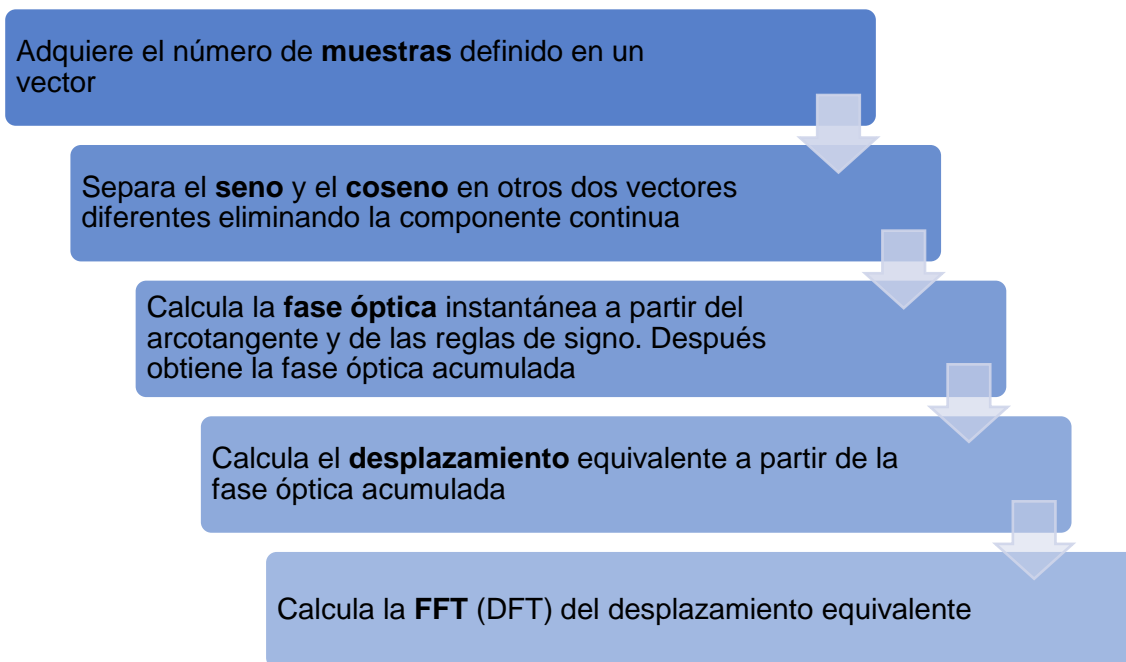


Figura 41. Operaciones y procesamiento de la señal de salida

Las conexiones del microcontrolador con la PCB (circuito electrónico de acondicionamiento) y el ordenador se definen en la **Figura 42**.

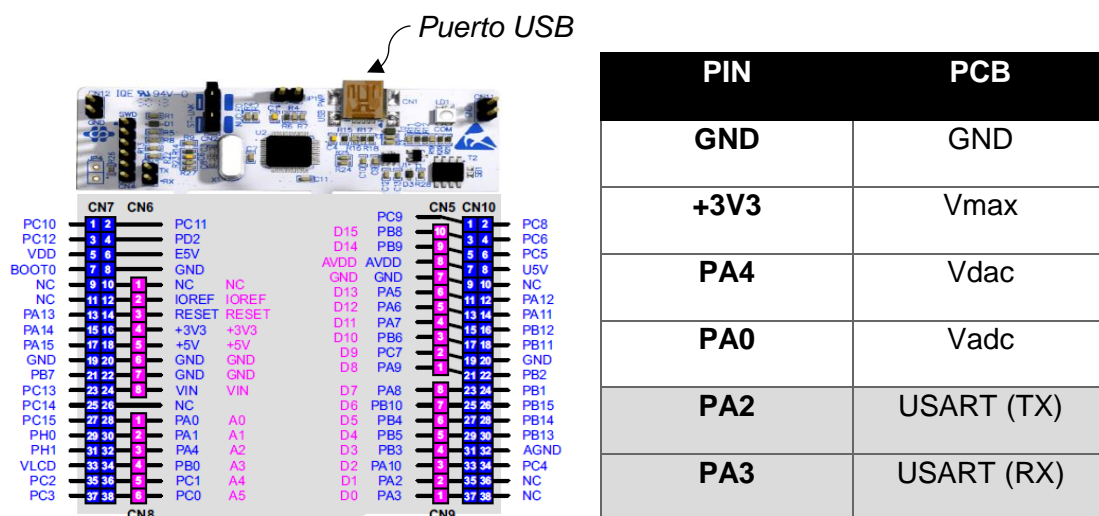


Figura 42. Conexiones del microcontrolador utilizadas [18]

Las dos últimas conexiones (USART) solo se realizan si no se van a transmitir los datos al ordenador mediante el puerto USB incorporado en la placa de desarrollo en ST-LINK (kit de depuración y programación integrado en la placa de desarrollo) o si se implementa el microcontrolador por separado o sin ST-LINK. En su lugar puede comunicarse con otro dispositivo mediante estas dos conexiones.

4.3. APLICACIÓN INFORMÁTICA

El diagrama de bloques de este subsistema y su relación con el resto se muestra en la **Figura 43**.

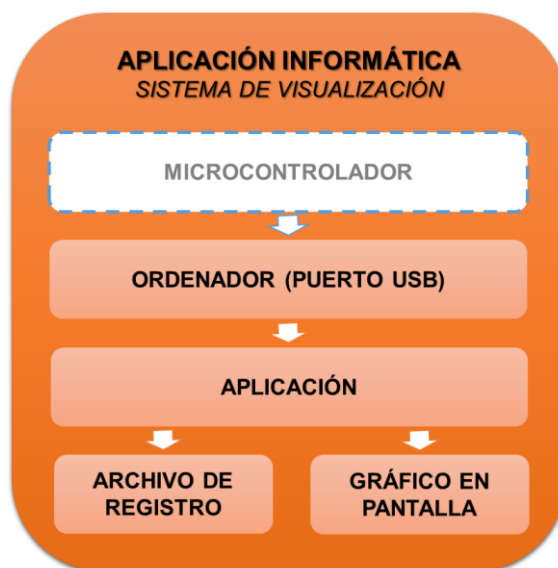


Figura 43. Diagrama de bloques de la aplicación informática (visualización)

El objetivo de este subsistema es el control y la visualización de los resultados del sistema por parte del usuario a través de una interfaz gráfica intuitiva y fácil de usar. Además, posibilitará el registro de las mediciones realizadas.

4.3.1. TECNOLOGÍAS UTILIZADAS Y DESARROLLO DE LA APLICACIÓN

Para el desarrollo de la aplicación se han utilizado tecnologías que permitan la portabilidad de la aplicación informática en diferentes sistemas operativos, buscando también la adaptabilidad en entornos futuros basados en aplicaciones web.

Por ello, se ha escogido trabajar con Electron⁷. Se trata de un *framework* que permite crear aplicaciones híbridas a partir de tecnologías web existentes: JavaScript, HTML y CSS. Hace uso de Chromium (explorador web) y Node.js (entorno de ejecución) como componentes esenciales para el funcionamiento de la aplicación. En este contexto y a grandes rasgos, la aplicación se desarrolla como una aplicación web normal, pero con la posibilidad de aprovechar los recursos específicos del sistema en el que se ejecuta (incluyendo los puertos USB para la comunicación serie). Una vez desarrollada la aplicación esta es empaquetada y se compila para cada uno de los sistemas operativos en los que va a ejecutarse. La **Figura 44** compara las características entre una aplicación nativa, una aplicación híbrida y una aplicación web.

⁷ Más información: <https://electronjs.org/>

NATIVE vs. WEB vs. HYBRID: 7 FACTORS OF COMPARISON				KEY	CON	PRO	NEUTRAL
	NATIVE	HYBRID	WEB				
COST	Commonly the highest of the three choices if developing for multiple platforms	Similar to pure web costs, but extra skills are required for hybrid tools	Lowest cost due to single codebase and common skillset				
CODE REUSABILITY/PORTABILITY	Code for one platform only works for that platform	Most hybrid tools will enable portability of a single codebase to the major mobile platforms	Browser compatibility and performance are the only concerns				
DEVICE ACCESS	Platform SDK enables access to all device APIs	Many device APIs closed to web apps can be accessed, depending on the tool	Only a few device APIs like geolocation can be accessed, but the number is growing				
UI CONSISTENCY	Platform comes with familiar, original UI components	UI frameworks can achieve a fairly native look	UI frameworks can achieve a fairly native look				
DISTRIBUTION	App stores provide marketing benefits, but also have requirements and restrictions	App stores provide marketing benefits, but also have requirements and restrictions	No restrictions to launch, but there are no app store benefits				
PERFORMANCE	Native code has direct access to platform functionality, resulting in better performance	For complex apps, the abstraction layers often prevent native-like performance	Performance is based on browser and network connection				
MONETIZATION	More monetization opportunities, but stores take a percentage	More monetization opportunities, but stores take a percentage	No store commissions or setup costs, but there are few monetization methods				

Figura 44. Comparación entre los distintos tipos de aplicaciones en la actualidad [20]

Cada una presenta sus ventajas y desventajas, enfocándose en un uso final concreto. En este caso se ha escogido desarrollar una aplicación híbrida principalmente por las siguientes razones: posibilidad de hacer portable la aplicación en diferentes plataformas o sistemas operativos; la no necesidad de aprovechar todo el potencial del sistema en el que se ejecuta (no se requiere procesamiento complejo de información); y la posibilidad de convertirla en una aplicación web pura en un futuro, accesible desde cualquier lugar.

Además de Electron, también se ha utilizado Angular⁸, otro *framework* que permite crear aplicaciones web multiplataforma en TypeScript (que aporta algunas ventajas sobre el lenguaje JavaScript) y proporciona la posibilidad de incorporar elementos gráficos de diseño de Android (botones, diálogos, selectores, etc.) en HTML, resultando finalmente en una interfaz gráfica sencilla y atractiva para el usuario.

De esta manera, el código fuente de la aplicación, recogido y comentado en el **ANEXO IV: Código fuente de la aplicación informática**, consta de dos partes principales. La primera, escrita en TypeScript, se encarga de la lógica de la aplicación. La segunda, escrita en HTML, se encarga de la interfaz gráfica de la aplicación. Para el desarrollo de la aplicación se ha utilizado el entorno de desarrollo integrado Visual Studio Code⁹.

4.3.2. USO DE LA APLICACIÓN Y VISUALIZACIÓN DE LOS RESULTADOS

La aplicación consta de tres pantallas distribuidas en tres pestañas: “Configuración”, “Medición” e “Historial”. En los siguientes apartados se expone la utilidad de cada una de ellas.

4.3.2.1. CONFIGURACIÓN

En esta pestaña se configuran los parámetros del sistema, enviando estos al microcontrolador a través de la comunicación serie (**Figura 45**).

⁸ Más información: <https://angular.io/>

⁹ Más información: <https://code.visualstudio.com/>



Figura 45. Pestaña “Configuración” de la aplicación informática

Los pasos que deben seguirse para configurar el sistema son:

1. **Conexión:** el usuario selecciona el puerto COM correspondiente al puerto USB al que se encuentra conectado el microcontrolador, es decir, se establece la comunicación serie mediante USART (**Figura 46**). Una vez conectado se restablecen los parámetros del sistema a sus valores predeterminados.



Figura 46. Conexión del microcontrolador

2. **Parámetros:** el usuario selecciona los parámetros de muestreo (**Figura 47**): “lambda/n” (longitud de onda del láser dividida por el índice de refracción de la fibra óptica), “Fmodulación” (frecuencia de la señal de modulación) y “Nmuestras” (número de muestras a adquirir considerando máximos y mínimos: 64, 128, 256, 512, 1024, 2048, 4096 o 8192).

Conexión — 2 Parámetros — 3 Adquisición — 4 Calibración

Parámetros de muestreo

Selecciona la lambda del láser, la frecuencia de modulación y el número de muestras de adquisición

Lambda/n = 633 nm

Fmodulación = 20000 Hz

Nmuestras = 2048

Anterior Siguiente

Figura 47. Parámetros de muestreo

3. **Adquisición:** el usuario modifica el rango de entrada en el ADC del microcontrolador ajustando el potenciómetro “1” de la PCB hasta conseguir un rango de adquisición óptimo. Para ello se siguen las indicaciones de la **Figura 48** cada vez que se hace clic en “Rango”.

Conexión — Parámetros — 3 Adquisición — 4 Calibración

Ajuste de ganancia en adquisición

Haz clic en *Rango* y sigue las instrucciones hasta conseguir un rango óptimo

Rango indefinido...

Rango

Anterior Siguiente

Rango indefinido...
Ajusta el rango girando el potenciómetro 1
¡Rango óptimo!

Figura 48. Ajuste de ganancia en adquisición

4. **Calibración:** el usuario calibra el sistema automáticamente o manualmente modificando la amplitud de modulación utilizando los controles disponibles (**Figura 49**). El potenciómetro “2” de la PCB debe ajustarse para proporcionar suficiente ganancia y posibilitar la calibración automática.

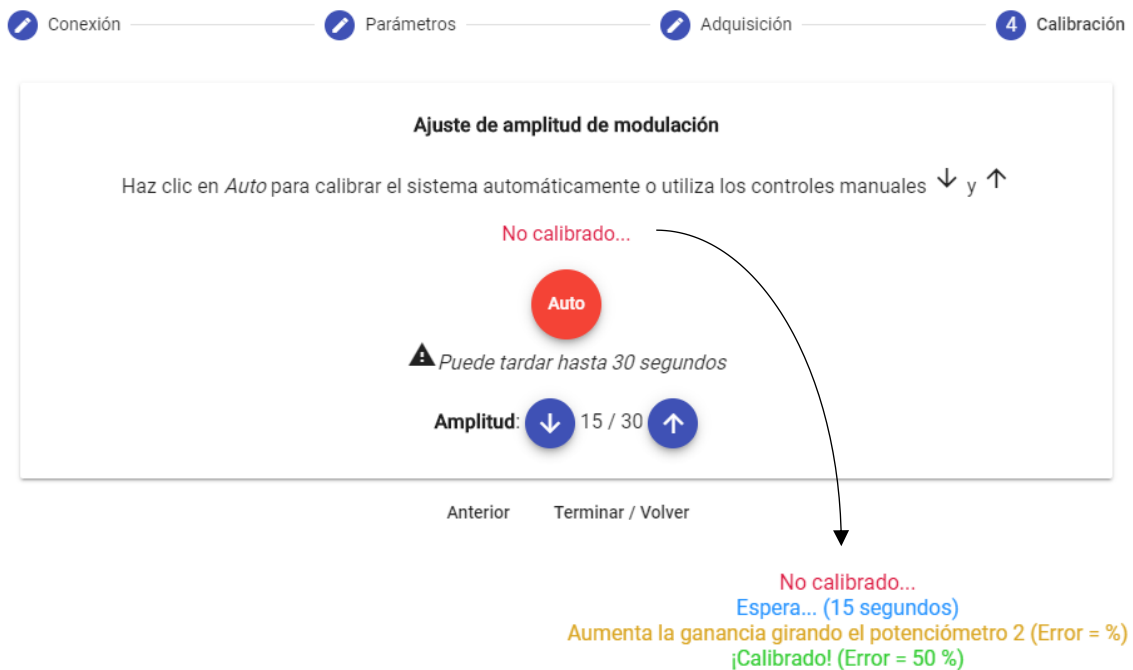


Figura 49. Ajuste de amplitud de modulación

4.3.2.2. MEDICIÓN

Una vez configurado el sistema, se pueden realizar mediciones a través de los controles de la pestaña “Medición” (**Figura 50**).



Figura 50. Pestaña “Medición” de la aplicación informática

Las mediciones pueden hacerse de forma manual (única medición en el instante en el que se solicita realizarla) o automática (medición continua cada cierto intervalo de tiempo). Con la medición automática se puede monitorizar un sistema o maquinaria continuamente durante largos periodos de tiempo sin intervención directa del usuario.

Por un lado, para la medición manual la pestaña dispone de tres botones dedicados:

- **Adquirir:** solicita al microcontrolador realizar una medición en el instante en el que se pulsa (con los parámetros especificados en la pestaña “Configuración”)
- **Detalles:** muestra cualquiera de los gráficos intermedios en función del tiempo de la última medición realizada (**Figura 51**): muestras (muestras adquiridas en bruto), seno (la primera señal de las dos en cuadratura), coseno (la segunda señal de las dos en cuadratura), fase (fase óptica acumulada) y desplazamiento (en función de la fase óptica acumulada y del instante inicial de la primera muestra).

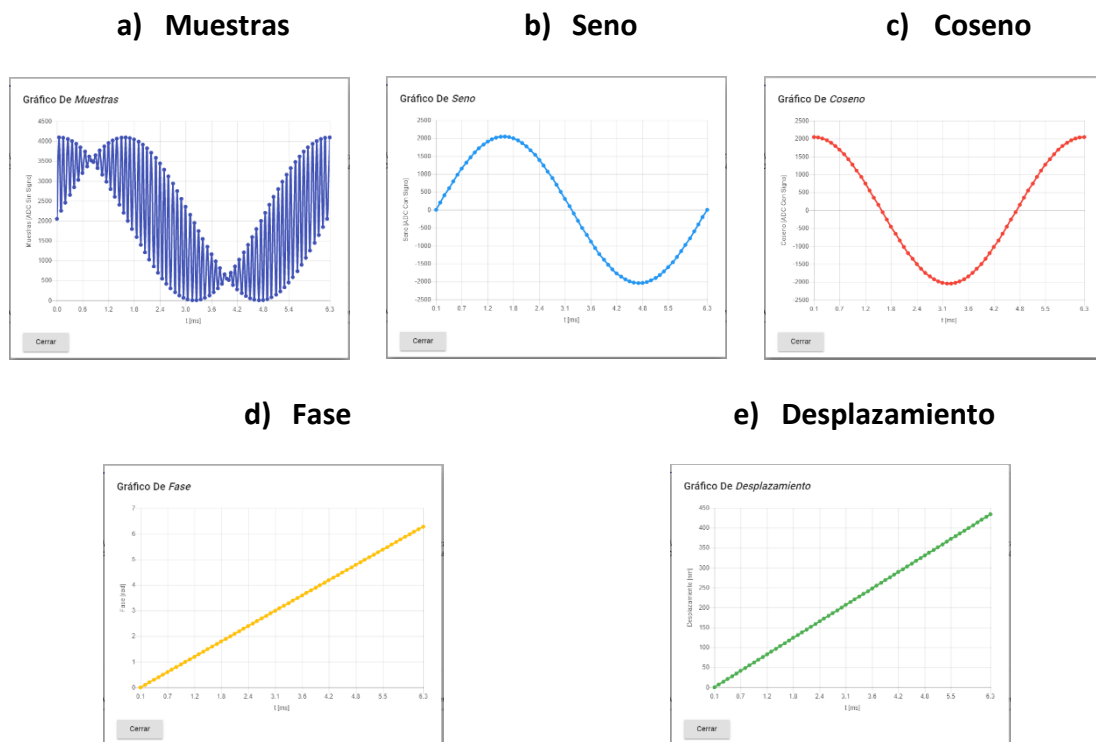


Figura 51. Gráficos intermedios (desplazamiento de fase constante de 0 a 2π radianes)

Para la visualización de los gráficos se ha utilizado una librería específica de JavaScript: Chart.js¹⁰, que permite configurar la estructura y el formato de los mismos.

- **Medida:** muestra el gráfico de la FFT de la última medición realizada, siendo posible seleccionar los puntos para obtener exactamente la frecuencia y el valor del desplazamiento en ese punto (**Figura 52**).

¹⁰ Más información: <https://www.chartjs.org/>

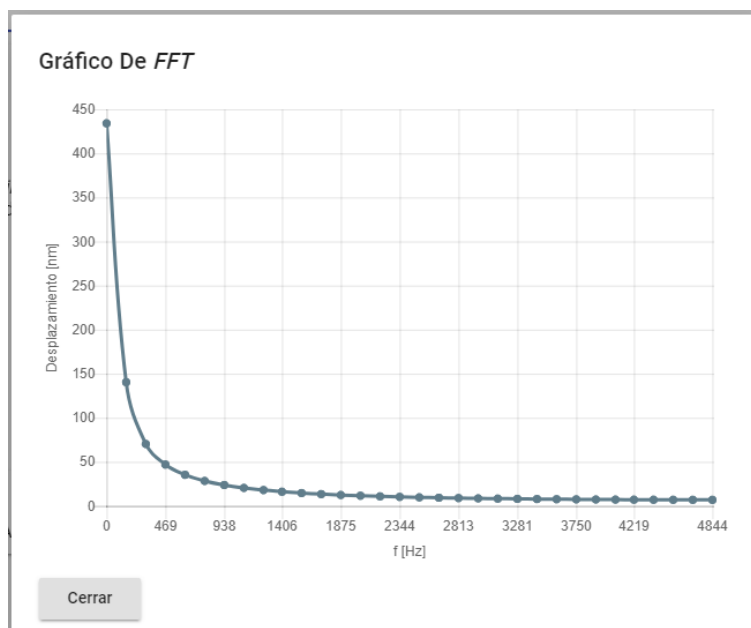


Figura 52. Gráfico de la FFT

Por otro lado, para adquirir automáticamente cada cierto intervalo de tiempo se debe configurar el tiempo y activar la adquisición programada mediante el selector y el interruptor de la **Figura 53**.

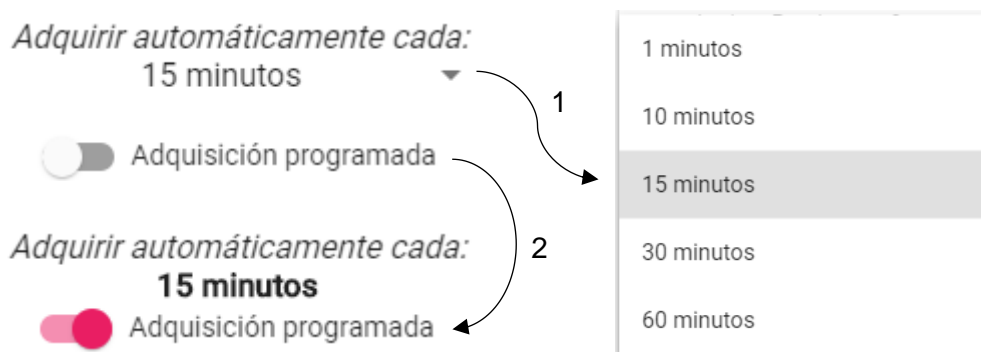


Figura 53. Configuración de la adquisición programada

4.3.2.3. HISTORIAL

Las mediciones realizadas durante la ejecución de la aplicación se almacenan temporalmente y se pueden volver a visualizar desde la pestaña “Historial” (**Figura 54**).

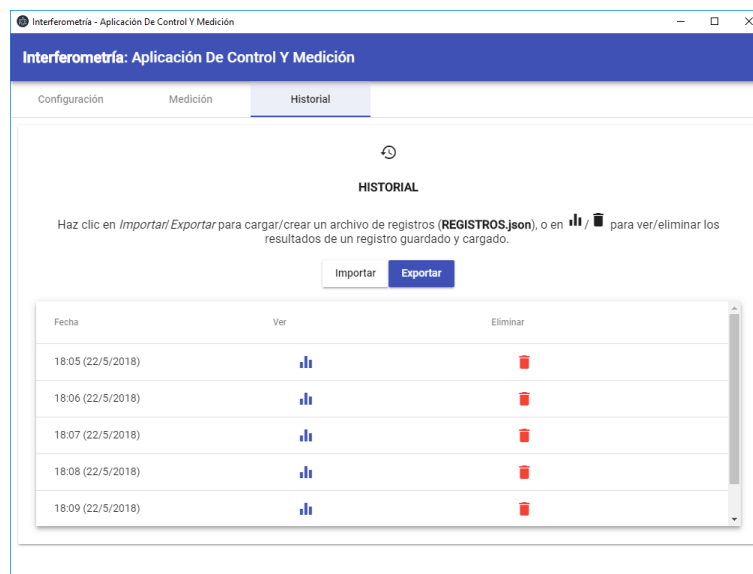


Figura 54. Pestaña “Historial” de la aplicación informática

En esta pestaña se puede volver a mostrar el gráfico de la FFT de una medición disponible o eliminarla mediante los botones correspondientes. También es posible exportar las mediciones guardadas temporalmente a un archivo o importar mediciones anteriores desde un archivo. El registro de las mediciones en un archivo se explica en el siguiente apartado.

4.3.3. REGISTRO DE LAS MEDICIONES

La aplicación permite guardar las medidas obtenidas (FFT) en un archivo, brindando la posibilidad de hacer un posterior análisis o procesamiento de los datos para otros fines como la mejora del control de calidad aplicando algoritmos de *Machine Learning*, por ejemplo.

El formato de los datos escogido es JSON, y por tanto la extensión del archivo exportado será “.json” (el nombre definido es “REGISTROS.json”). Este archivo tiene la ventaja de ser legible desde cualquier editor de texto, además de poder ser incorporado a una base de datos no relacional fácilmente. Los datos guardados de cada medición se muestran en la **Tabla 3**.

Tabla 3. Atributos de los datos guardados de cada medición

ATRIBUTO	DESCRIPCIÓN
Fecha	Fecha en la que se obtuvo la medida
Muestras	Número de muestras adquiridas
Frecuencia	Frecuencia de modulación
FFT	Datos de desplazamiento de la FFT

Cada medición será un elemento del vector guardado en el archivo. El aspecto de los registros en formato JSON en texto plano se muestra en la **Figura 55**.

```
[
  {
    "fecha": "18:30 (15/3/2018)",
    "muestras": 2048,
    "frecuencia": 20000,
    "fft": [
      120.669472,
      6.797237,
      ...,
      0.208548,
      0.201844
    ]
  },
  ...
]
```

Figura 55. Formato de los datos guardados en JSON

Las muestras y la frecuencia son dos parámetros que, además de proporcionar información sobre las condiciones de la medición, también son utilizados para reconstruir la FFT, ya que es necesario obtener los puntos en el eje frecuencial (f_i) a partir de la siguiente relación:

$$f_i = i \cdot \frac{f_{mod}}{\left(\frac{N}{2}\right)} \quad i = [0, N - 1] \quad (4.11)$$

donde f_{mod} es la frecuencia de modulación, N es el número de muestras guardado e i es el número de la muestra (de 0 a $N - 1$).

De esta manera, se evita guardar los valores del eje frecuencial para cada una de las medidas.

5. EJEMPLO DE APLICACIÓN

En este capítulo se realizan pruebas de funcionamiento del sistema completo, haciendo uso del mismo para un caso concreto: medición de las vibraciones del núcleo de un transformador de baja potencia.

El primer paso es encender la fuente del láser una vez instalado el sistema óptico y después conectar el microcontrolador al ordenador. Antes de controlar el sistema a través de la aplicación se va a comprobar la comunicación serie entre el ordenador y el microcontrolador. Para ello se va a utilizar el programa Termite¹¹ que permite enviar comandos en forma de texto (codificación ASCII) y recibir la respuesta del dispositivo mediante comunicación serie. En la **Figura 56** se muestra el registro de algunos comandos y las respuestas recibidas del microcontrolador (en azul y verde respectivamente).

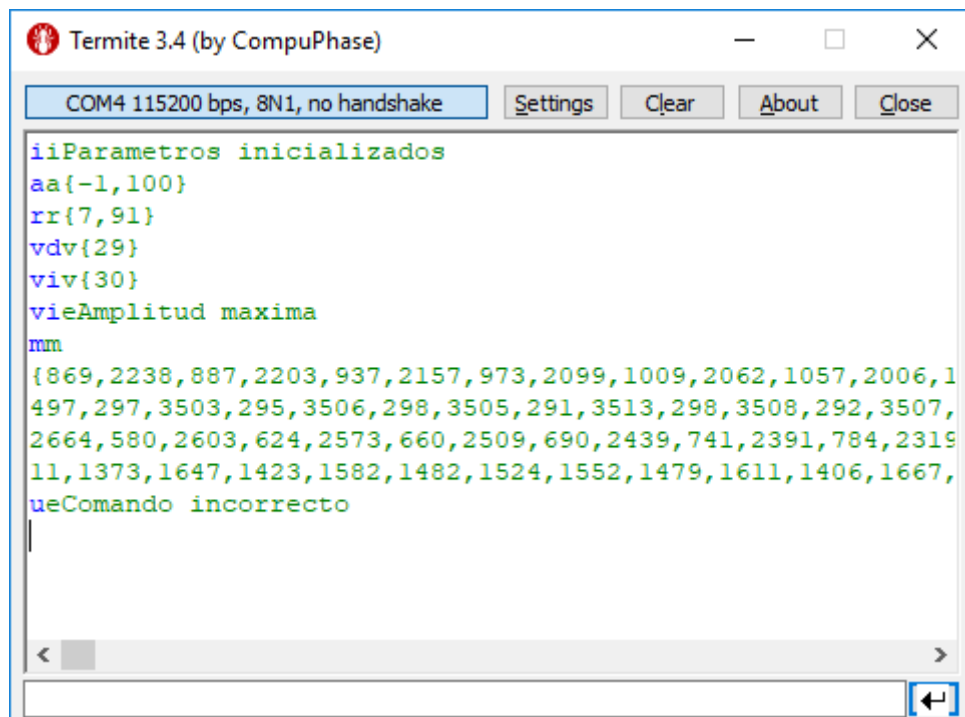


Figura 56. Comunicación serie entre el microcontrolador (verde) y el ordenador (azul)

Una vez se ha verificado la comunicación entre ambos dispositivos (opcional), se ha comprobado que se produce interferencia a la salida del interferómetro (pudiendo regular la polarización de los haces de luz mediante el control de polarización) y se han realizado las conexiones eléctricas necesarias, se accede a la aplicación informática y se establece la comunicación como se indica en la **Figura 45**. A continuación, se siguen los pasos guiados en la pestaña “Configuración”. En este caso se ha escogido la siguiente configuración:

- **Lambda/n:** 434 nm
- **Fmodulación:** 20000 Hz

¹¹ Más información: https://www.compuphase.com/software_termite.htm

- **Nmuestras:** 2048

Después de configurar la ganancia en adquisición y de calibrarse el sistema automáticamente, se puede observar en la **Figura 57** la señal de modulación aplicada al PZT (resultado de acondicionar la señal de modulación proporcionada por el microcontrolador) junto a la señal de salida acondicionada para ser adquirida en un rango de 0 a 3.3 V (una vez ajustados los potenciómetros del circuito de acondicionamiento).

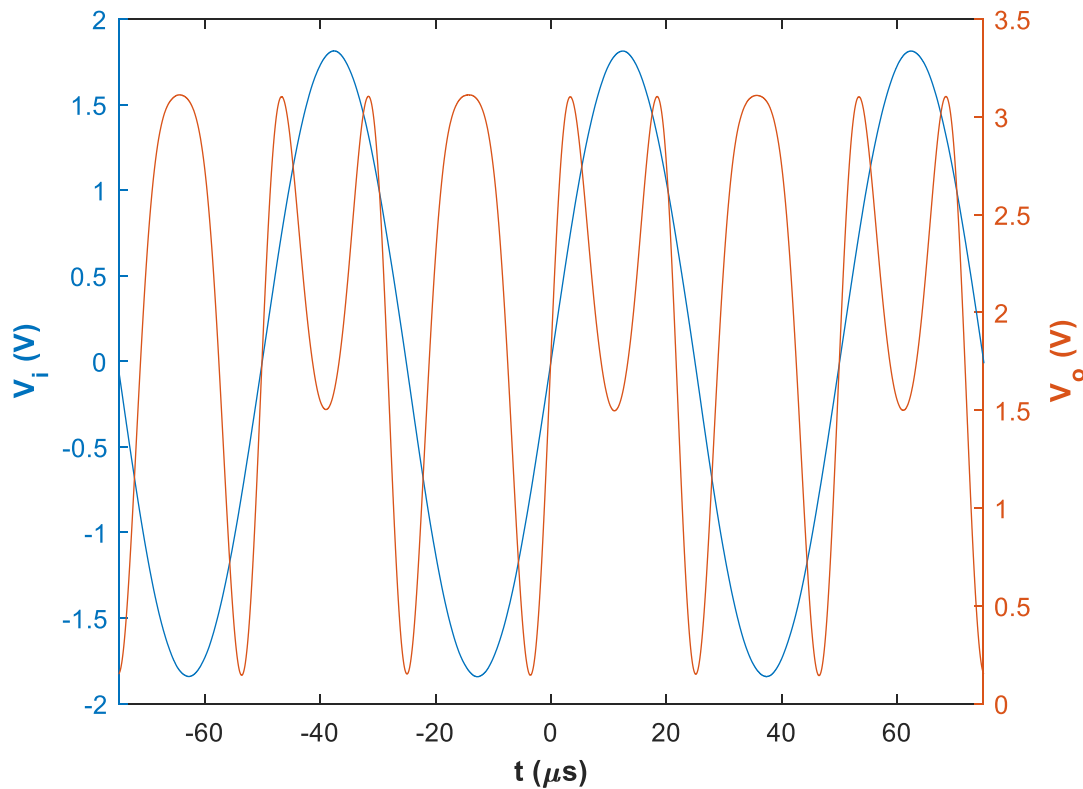
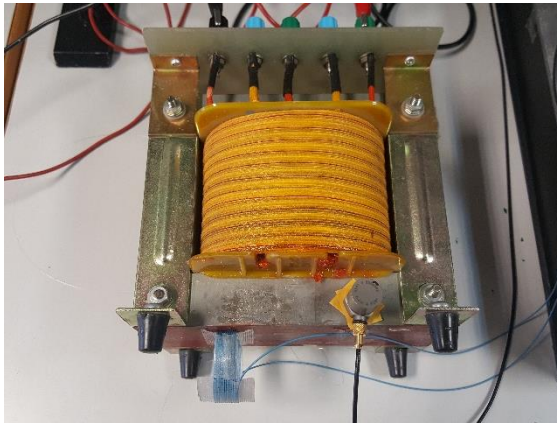


Figura 57. Señales de modulación V_i (acondicionada de la proporcionada por el microcontrolador y que se aplica al PZT) y de salida del interferómetro V_o (acondicionada para ser adquirida por el microcontrolador)

A continuación, se procede a obtener la medida deseada. Para ello la fibra óptica debe estar en contacto con el objeto del que se desea medir las vibraciones. Las vibraciones en el núcleo del transformador de baja potencia tienen origen por el efecto de las fuerzas de magnetostricción [21]. Estas vibraciones son de muy baja intensidad, por lo que solo es posible detectarlas eficazmente (con el menor ruido) si se tiene una muy buena sensibilidad en la medida. El armónico fundamental de estas vibraciones se encuentra a 100 Hz. En este caso, para detectarlas se va a magnificar el efecto de la deformación producida en la fibra óptica enrollando varias veces la fibra óptica para aumentar el área de contacto antes de colocarla en el núcleo del transformador como se observa en la **Figura 58** (mismo recurso que se utilizó para la generación de la señal de modulación con menor tensión aplicada al PZT).

a) Transformador de baja potencia



b) Detalle de la fibra óptica

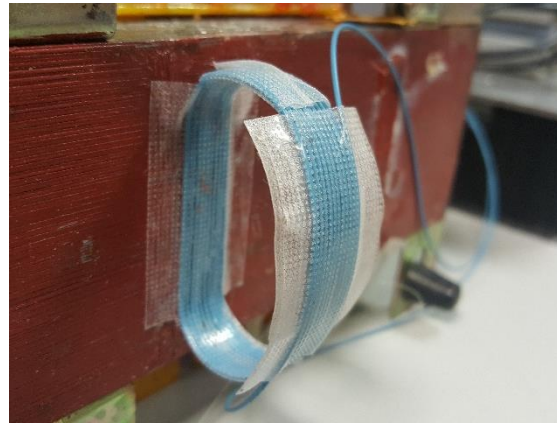


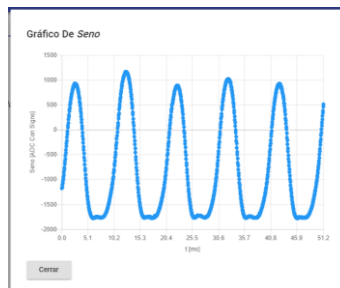
Figura 58. Transformador eléctrico de baja potencia (a) con la fibra óptica enrollada adherida al núcleo del transformador (b)

Una vez colocada la fibra óptica, se procede a realizar la medición en la pestaña “Medición” haciendo clic en el botón “Adquirir”. La tensión del transformador es $V_{tra} = 230\text{ V}$ durante el tiempo de medición. Los gráficos intermedios obtenidos se muestran en la **Figura 59**.

a) Muestras



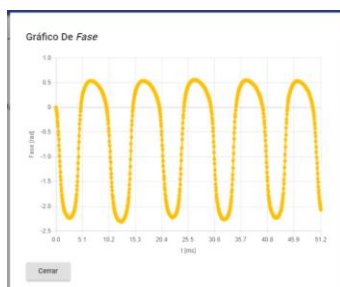
b) Seno



c) Coseno



d) Fase



f) Desplazamiento

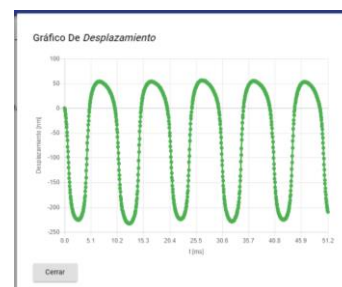


Figura 59. Gráficos intermedios obtenidos en la medición del transformador

La gráfica de la FFT obtenida se muestra en la **Figura 60**.

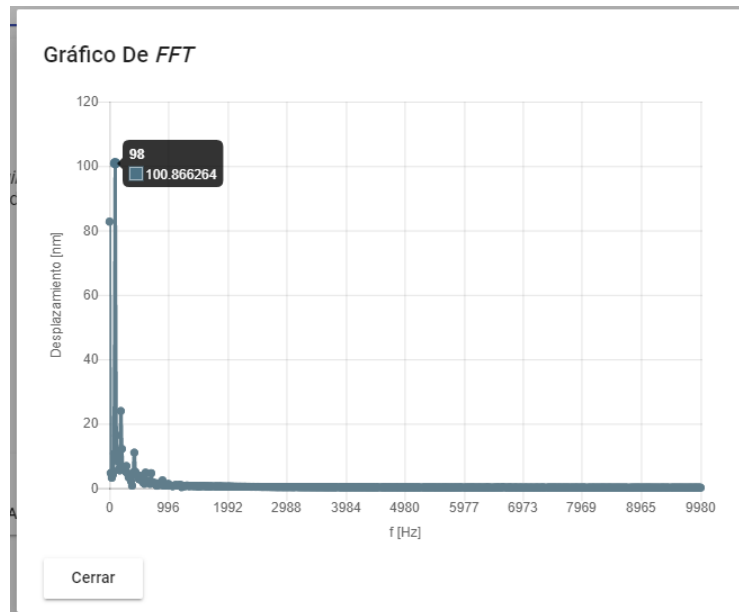


Figura 60. Gráfico de la FFT de la medición del transformador

Se puede observar que se obtiene el armónico fundamental a aproximadamente 100 Hz. Para comprobar los resultados obtenidos se ha realizado el mismo experimento de medición de las vibraciones, pero manualmente. Para ello se utilizó un osciloscopio digital¹² que permite observar la señal de salida del interferómetro y la señal de modulación al mismo tiempo, así como registrar la evolución temporal de ambas y procesarlas posteriormente.

El primer paso antes de utilizar el interferómetro es calibrar la fase óptica de modulación, que se hace una sola vez para obtener el número de mediciones necesarias de cualquier objeto. Como ya se ha mencionado, existen múltiples posibilidades para hacerlo. No obstante, la facilidad de calibración para una amplitud de pico de la fase $\phi_{mod_0} = \pi + \frac{\pi}{4}$ radianes, no siendo necesario conocer la amplitud máxima y mínima a la salida pudiendo observar dicho desfase en cualquier instante, hace de esta la opción ideal para realizar la calibración manualmente (y automáticamente). En la **Figura 61** se representan ambas señales en el osciloscopio cuando el sistema se ha calibrado.

¹² Se ha utilizado el osciloscopio digital *Cleverscope CS328A*. Hoja de características: <https://cleverscope.com/products/CS328A>

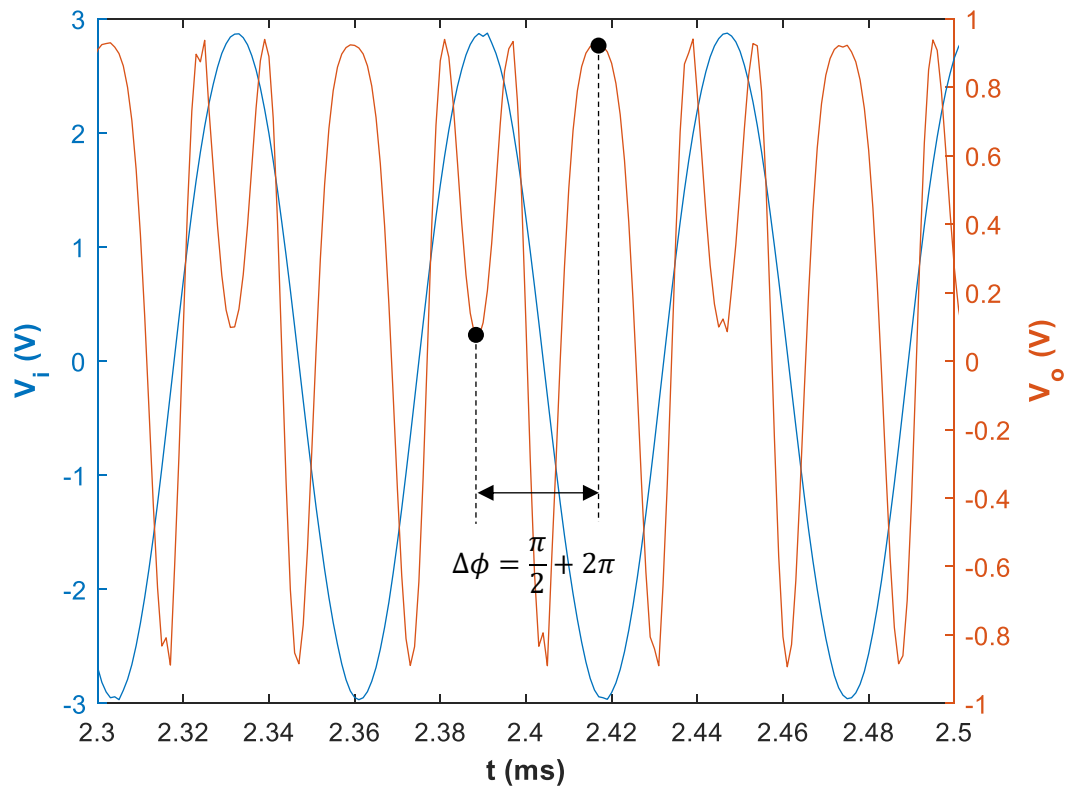


Figura 61. Comprobación de la calibración manual del interferómetro (las dos señales en los máximos y mínimos se encuentran en cuadratura)

En la adquisición de las señales, el transformador se encontraba a la misma tensión de alimentación $V_{tra} = 230 \text{ V}$. La **Figura 62** muestra la señal de modulación, la de salida y la reconstruida junto al espectro de frecuencias obtenido después de procesar $N = 1750$ muestras de fase instantánea, realizando los procedimientos descritos en el Capítulo 3.

Destacar que en esta comprobación manual se deben obtener los valores de ambas señales (la de modulación y la de salida) y con una frecuencia de muestreo bastante mayor para procesar los datos correctamente, no pudiendo seleccionar de otra manera los máximos y mínimos. Esta es una de las principales ventajas al usar el sistema completo junto a los bloques de acondicionamiento, procesamiento y visualización, ya que se reduce considerablemente el almacenamiento requerido para los datos adquiridos.

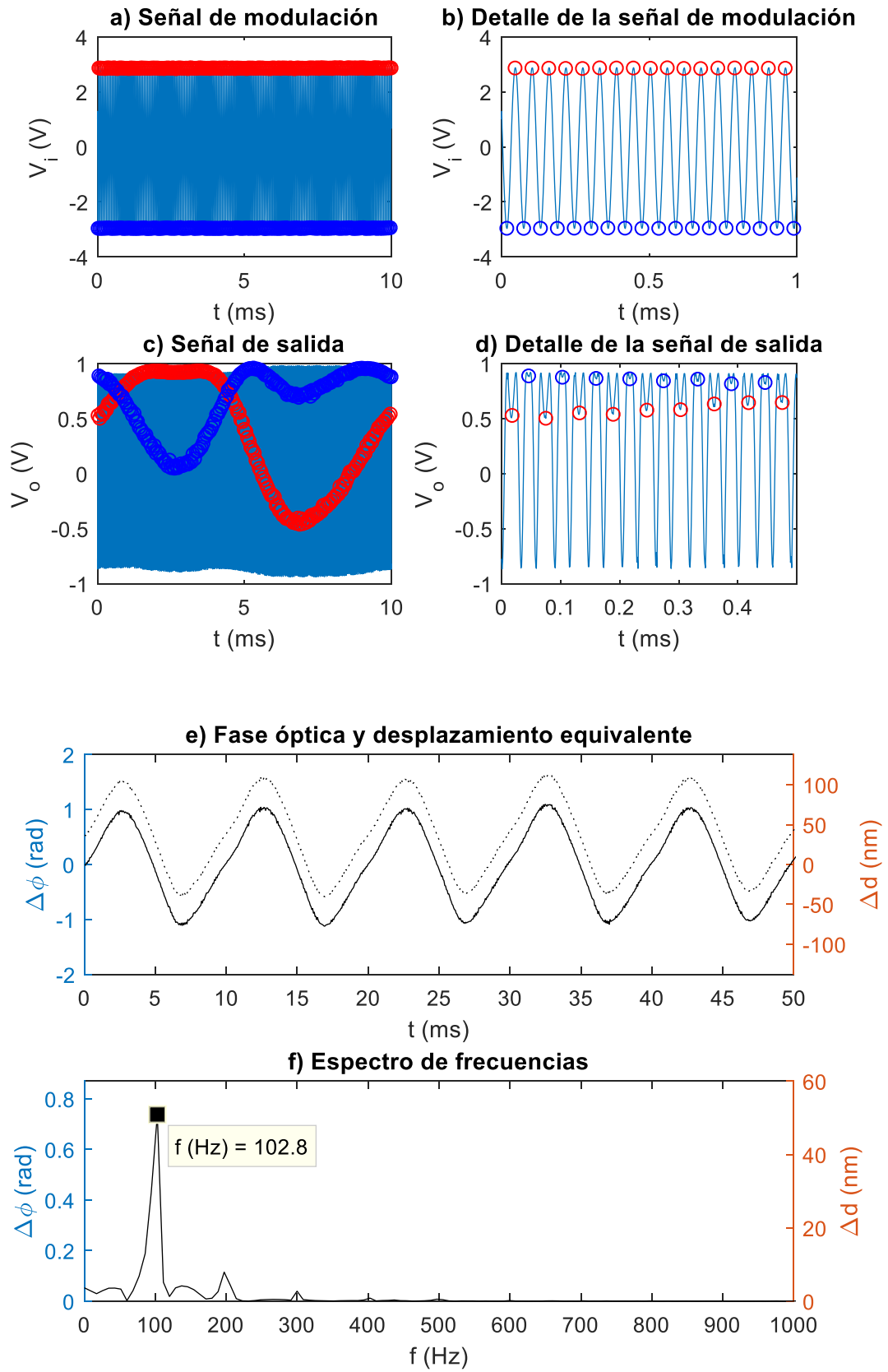


Figura 62. Resultados obtenidos manualmente en la medición de las vibraciones del transformador

Como se puede observar, los resultados son prácticamente idénticos a los obtenidos con los bloques de acondicionamiento, procesamiento y visualización.

Por último, para demostrar las ventajas de utilizar el sensor de fibra óptica es interesante comparar el resultado anterior con el que se obtiene mediante el uso de un acelerómetro comercial. El acelerómetro utilizado presenta una sensibilidad de $10 \frac{V}{m/s^2}$. Conociendo este dato se puede obtener el desplazamiento integrando dos veces la aceleración registrada en el tiempo, representándose su evolución temporal y el espectro de frecuencias en la **Figura 63**. La señal temporal del desplazamiento obtenido (**Figura 63c**) ha sido filtrada para eliminar las componentes de baja frecuencia.

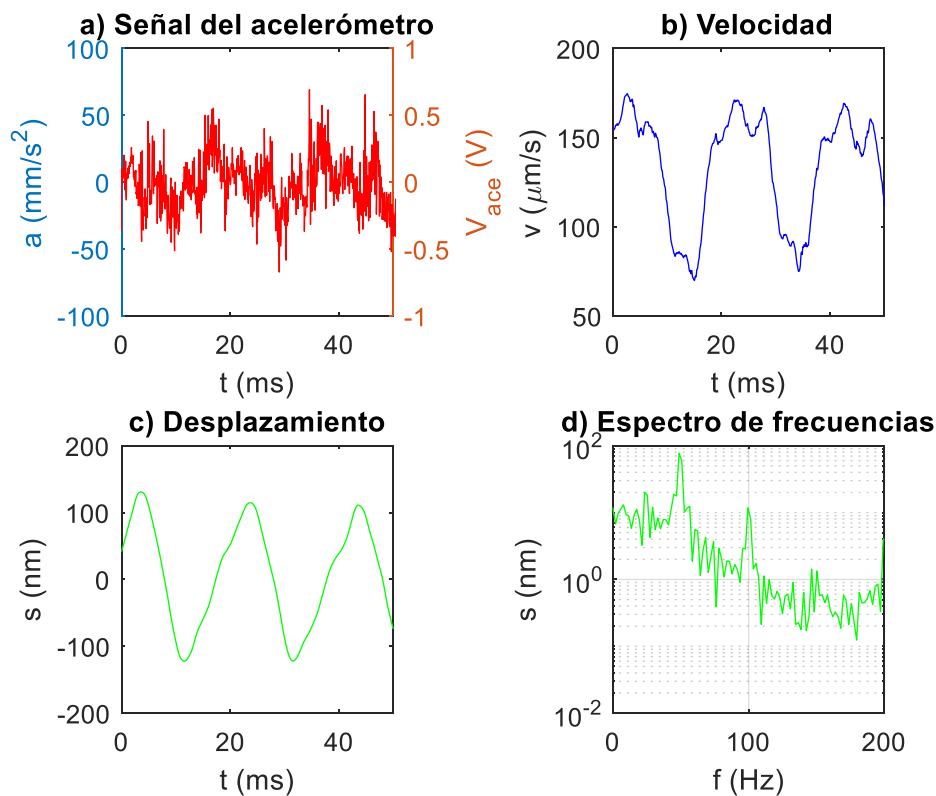


Figura 63. Resultados obtenidos del acelerómetro en la medición de las vibraciones del transformador

Se puede observar como en el espectro de frecuencias del acelerómetro se muestra un mayor número de armónicos que en interferómetro. Estos pueden ser originados por diferentes causas como la interferencia electromagnética, por ejemplo. Destaca el armónico a 50 Hz (frecuencia de red), que se detecta con un valor mayor que el de las vibraciones a 100 Hz. Este armónico aparece mucho más atenuado en el espectro de frecuencias del sensor de fibra óptica, demostrando su inmunidad frente a este tipo de influencias.

6. CONCLUSIÓN Y LÍNEAS FUTURAS

En este proyecto se ha diseñado un sistema completo de interferometría láser basado en fibra óptica con una técnica de detección síncrona de la fase óptica, aplicándose finalmente a la medición de vibraciones en máquinas.

En primer lugar, la técnica desarrollada resuelve un problema común en la interferometría láser: obtener la fase óptica. Frente a otras técnicas presenta algunas ventajas esenciales: procesamiento digital relativamente sencillo para la reconstrucción de la señal; mejora del rango dinámico de medida; mejora de la sensibilidad; elimina la ambigüedad de signo; no necesita un punto de trabajo fijo; y es independiente de la amplitud de la intensidad óptica a la salida. A partir de los resultados obtenidos se deduce que la técnica propuesta es una alternativa posible frente a otras técnicas complejas, a pesar del pequeño error cometido (prácticamente despreciable en muchos casos) y de las limitaciones en frecuencia al usar un PZT. La simplicidad de la instalación óptica, los bajos requerimientos y coste, y la posible escalabilidad hacen esta una opción perfectamente válida en numerosas situaciones.

En cuanto al diseño e implementación del sistema completo, se ha logrado una interfaz final para el usuario intuitiva y atractiva que permite la rapidez de uso del sistema y además un registro de las mediciones para un posterior análisis. Además, los sistemas de acondicionamiento y control diseñados permiten su posterior portabilidad a otra instalación de interferencia óptica de diferentes características.

A pesar de haberse cumplido los objetivos propuestos, en el trabajo se identifican varias líneas futuras o posibles mejoras que podrían llevarse a cabo.

La técnica de demodulación de la fase óptica se ha aplicado para la medición de vibraciones y el trabajo se ha enfocado en ofrecer una solución al respecto. No obstante, la técnica no está limitada únicamente a ello y puede aplicarse a la medida de otras magnitudes físicas basadas en la obtención de la fase óptica. Un ejemplo es la temperatura que podría extraerse acotando las frecuencias de las vibraciones (aunque sería necesario un punto de referencia para una medida absoluta). También puede aplicarse a distintos esquemas de interferencia para la medida sin contacto y la modulación puede realizarse sobre el láser a frecuencias del orden de megahercios a través de la corriente de inyección del láser para detectar ultrasonidos (superando la barrera de frecuencias delimitada por el PZT que se comporta como un condensador limitando la velocidad de respuesta en la carga y descarga del mismo). De igual manera, para obtener medidas a lo largo de la fibra óptica se podría utilizar una red de Bragg en fibra (FBG, *Fiber Bragg Grating*) que permitiría incluso multiplexar varios sensores de fibra óptica para obtener la medida deseada en diversos puntos: temperatura, deformación, presión, aceleración, etc. En este caso el interferómetro se utilizaría para interrogar el sensor o conjunto de sensores.

Por último, la implementación de los bloques de acondicionamiento, control y adquisición, y la aplicación informática también pueden mejorarse. El sistema presentado no requiere apenas la intervención humana para su calibración y puesta en marcha. Sin embargo, en la solución propuesta se utilizan potenciómetros manuales para realizar una acción que podría también ser automatizada por el microcontrolador mediante el uso de potenciómetros digitales. En el

procesamiento se podría llegar a corregir el pequeño error originado por el desfase de las señales en cuadratura o posibilitar las mediciones continuas en tiempo real si se optimizaran suficientemente los cálculos. En cuanto a la visualización de los resultados, el acondicionamiento, el control y la adquisición podrían integrarse conjuntamente, y también podría añadirse un módulo *Bluetooth* o *WiFi* convirtiéndolo en un producto compacto y sin cable de comunicación, enviando los datos inalámbricamente a uno o varios dispositivos al mismo tiempo. Además, mediante las tecnologías escogidas para el desarrollo de la aplicación informática sería posible portarla a diferentes plataformas móviles con poco esfuerzo y adaptarla al nuevo modo de comunicación.

Este trabajo va a ser presentado en el 25º Seminario Anual de Automática, Electrónica industrial e Instrumentación (SAAEI) que tiene lugar los días 4-6 de Julio de 2018 en Barcelona. En el ***ANEXO V: Resumen presentado en el SAAEI 2018*** se incluye el resumen presentado.

7. MARCO REGULADOR

En este apartado se realiza un breve análisis sobre los aspectos legales y técnicos más importantes del proyecto.

En primer lugar, se deben tener presentes los riesgos del sistema desarrollado y estándares técnicos aplicables. El riesgo más notable es el uso de un láser, por lo que el trabajador o los trabajadores responsables deben cumplir lo dispuesto en el Real Decreto 486/2010 sobre la protección de la salud y la seguridad de los trabajadores contra los riesgos relacionados con la exposición a radiaciones ópticas artificiales. La norma UNE-EN 60825-1:2008, referenciada en el Real Decreto anterior, clasifica los láseres en función de su peligrosidad, expone los requisitos mínimos de seguridad en la fabricación de los mismos, su etiquetado, y las medidas de control y protección que se deben adoptar. En cuanto a posibles estándares técnicos relacionados con el ejemplo de aplicación, se puede destacar el estándar IEEE 1415 [22] que proporciona información sobre pruebas y análisis de fallos sobre maquinaria de inducción.

En segundo lugar, en cuanto a la posible comercialización o patentabilidad del sistema desarrollado en un futuro si se transformara en un producto, hay dos consideraciones importantes. Por un lado, el sistema implementado es de carácter electrónico. Por ello, para la venta del producto en elEEE (Espacio Económico Europeo) debería obtener antes el marcado CE (Conformidad Europea), que verifica que el producto se ha evaluado y cumple los requisitos de seguridad, sanidad y protección del medio ambiente exigidos por la UE (Unión Europea). Por otro lado, los programas utilizados están sujetos a su propia licencia. En concreto, el código fuente desarrollado para el microcontrolador debe usarse y redistribuirse en base a las condiciones descritas por la empresa comercializadora del software de programación, que se encuentran en la cabecera del mismo (**ANEXO II: Código fuente del microcontrolador**). Actualmente existen patentes relacionadas con el trabajo realizado: sistema de interferometría en cuadratura que proporciona también información sobre la dirección del desplazamiento de la fase [23] y modulador de desplazamiento de fase en cuadratura mejorado utilizado para comunicaciones [24].

Por último, para el desarrollo del proyecto se han utilizado tres lenguajes de programación. Todos los códigos fuente desarrollados, el nombre de los ficheros y su extensión cumplen las normas de estilo propuestas por cada organización desarrolladora de las librerías, software o hardware. Además, se emplean estándares recientes de cada uno de los lenguajes de programación maximizando de esta manera la compatibilidad entre distintos compiladores y su portabilidad. El código en C se basa en el estándar C11 (ISO/IEC 9899:2011). El código en TypeScript se basa en el estándar ECMAScript 2015 (ISO/IEC 16262:2015). El código en HTML se basa en el estándar HTML5 (ISO/IEC 15445).

8. ENTORNO SOCIOECONÓMICO

8.1. PRESUPUESTO

En este apartado se estiman los costes de desarrollo e implementación del sistema presentado, recogidos en detalle y clasificados por categorías en la **Tabla 4**. Para la elaboración del presupuesto se han tenido en cuenta los siguientes aspectos:

- La amortización de los equipos y software es lineal y con valor residual nulo, indicándose en la descripción si corresponde. El precio se calcula en base a la duración del trabajo del ingeniero (estimado en 3 meses).
- Los programas *KiCad*, *Node.js*, *Microsoft Visual Studio Code* y *Termite* son software de código abierto sujeto a licencias de uso libre y gratuito, de igual manera que lo son las librerías utilizadas para el desarrollo de la aplicación informática. La licencia de *Cadence OrCAD Lite* es académica y sin coste.
- En las tablas: “U.M.” es Unidad de Medida; “P.U.” es Precio Unitario; y “P.T.” es Precio Total. No se aplican los impuestos territoriales correspondientes ni el transporte o los gastos de gestión. Tampoco se aplican otros costes indirectos.

Tabla 4. Presupuesto

CÓDIGO	U.M.	DESCRIPCIÓN	MEDICIÓN	P.U. (€)	P.T. (€)
CAPÍTULO 1: EQUIPOS Y SOFTWARE					
1.01	unidad	Interferómetro Mach-Zehnder. Interferómetro compacto integrado con dos entradas ópticas y dos salidas ópticas. Incluye fotodiodos amplificados balanceados para obtener la señal eléctrica de monitorización. <i>Amortización lineal con vida útil de 2 años y precio de adquisición de 1716 €</i>	1	214.50	214.50
1.02	unidad	Ordenador personal. Sistema operativo Microsoft Windows, procesador Intel x86-64, 50 GB de almacenamiento HDD/SSD y 8 GB de memoria RAM <i>Amortización lineal con vida útil de 5 años y precio de adquisición de 500 €</i>	1	25.00	25.00

CÓDIGO	U.M.	DESCRIPCIÓN	MEDICIÓN	P.U. (€)	P.T. (€)
1.03	unidad	Osciloscopio digital CS328A. Incluye licencia del software “Cleverscope4” para su control. <i>Amortización lineal con vida útil de 7 años y precio de adquisición de 1161 €</i>	1	41.46	41.46
1.04	licencia	MathWorks MATLAB. Licencia de uso académico individual. <i>Amortización lineal con duración de 1 año y precio de adquisición de 250 €</i>	1	62.50	62.50
TOTAL (CAPÍTULO 1):					343.46
CAPÍTULO 2: COSTES MATERIALES					
2.01	unidad	Actuador piezoeléctrico. Cilindro piezoeléctrico cerámico con frecuencia de resonancia a 30 kHz y dimensiones físicas (radio externo x radio interno x ancho): 35 x 31 x 20 mm	1	35.00	35.00
2.02	unidad	Cables. Paquete de 40 unidades macho-hembra, 40 unidades hembra-hembra y 40 unidades macho-macho compatibles con <i>proto board</i> .	1	7.09	7.09
2.03	unidad	Láser. He-Ne de 633 nm de longitud de onda (rojo) y 1 mW de potencia con acondicionamiento para fibra óptica y con fuente de alimentación para el mismo.	1	1257.00	1257.00
2.04	unidad	Microcontrolador STM32L152RE. Placa de desarrollo NUCLEO-L152RE con el microcontrolador incluido. Incluye licencia del software “Atollic TrueSTUDIO for STM32” para su programación.	1	11.80	11.80

CÓDIGO	U.M.	DESCRIPCIÓN	MEDICIÓN	P.U. (€)	P.T. (€)
2.05	unidad	Prototipo del circuito electrónico de acondicionamiento. Incluye los materiales y recursos necesarios para la fabricación de una placa de circuito impreso con los siguientes componentes: 1x regulador de tensión positiva LM7810; 1x regulador de tensión negativa LM7910; 1x circuito integrado con 4 amplificadores operacionales; 2x resistencia 4.7 kΩ; 4x resistencia 1 kΩ; 2x potenciómetro 10 kΩ; 2x condensador 100 nF; 2x condensador 10 μF; 1x conector SMA; 1x zócalo para circuito integrado (14 pines); 6x conector cable-placa	1	50.00	50.00
TOTAL (CAPÍTULO 2):					1360.89
CAPÍTULO 3: RECURSOS HUMANOS					
3.01	hora	Ingeniero con especialidad en electrónica. Nivel de experiencia profesional mínimo junior.	300	15.00	4500.00
TOTAL (CAPÍTULO 3):					4500.00
<u>TOTAL:</u>					6162.68

La **Tabla 5** resume el presupuesto anterior indicando la suma total.

Tabla 5. Resumen del presupuesto

CAPÍTULO	DESCRIPCIÓN	P.T. (€)
1	EQUIPOS Y SOFTWARE	343.46
2	COSTES MATERIALES	1360.89
3	RECURSOS HUMANOS	4500.00
<u>TOTAL:</u>		6204.35

El presupuesto total de este proyecto asciende a la cantidad de **SEIS MIL DOSCIENTOS CUATRO CON TREINTA Y CINCO EUROS**.

8.2. IMPACTO SOCIOECONÓMICO

La interferometría está ganando cada vez más importancia en el sector empresarial. Cada vez son más las aplicaciones en las que se necesita una precisión extrema en los parámetros físicos medidos y además se descubren nuevas posibilidades y aplicaciones de gran potencial.

Un campo importante, como ya se ha mencionado en el trabajo, es el mantenimiento y la monitorización de máquinas y equipos. Actualmente existe el Mantenimiento Predictivo (*Predictive Maintenance*) [25] afín al modelo de negocio basado en servicios de numerosas empresas, que engloba técnicas diseñadas para predecir posibles fallas o averías y determinar cuándo es necesario llevar a cabo operaciones de mantenimiento o reparaciones, reduciendo de esta manera los costes que se podrían derivar en diagnósticos. En este caso, los datos obtenidos de la FFT están preparados para almacenarse en una base de datos y pueden llegar a tratarse con algoritmos de *Machine Learning* para determinar fallas futuras de una máquina monitorizada.

También es interesante mencionar su posible aportación en el control de calidad, pudiendo reducir los costes derivados por productos defectuosos. La precisión de las magnitudes medidas mediante un interferómetro láser garantiza una fabricación de calidad satisfactoria, y la posibilidad de obtener resultados precisos en ensayos (composición o desperfectos en materiales, por ejemplo) de multitud de parámetros (deformación, vibraciones, humedad, dilatación térmica, etc.).

9. BIBLIOGRAFÍA Y REFERENCIAS

- [1] P. Hariharan, *Basics of interferometry*, 2nd ed. Ámsterdam, Países Bajos: Elsevier Academic Press, 2007.
- [2] “El espectro electromagnético”, *Cuaderno de Cultura Científica*. [En línea]. Disponible en: <https://culturacientifica.com/2016/08/16/el-espectro-electromagnetico/>. [Accedido: Jun. 2018]
- [3] “Primera detección de ondas gravitacionales por tres dispositivos”, *Europa Press*, 27-sep-2017. [En línea]. Disponible en: <http://www.europapress.es/ciencia/astronomia/noticia-primera-deteccion-ondas-gravitacionales-tres-dispositivos-20170927184623.html>. [Accedido: Jun. 2018]
- [4] “Simulation of Young's Interference Experiment via Geometric Ray Tracing in OpticStudio”, *Zemax*, 6-feb-2015. [En línea]. Disponible en: <http://customers.zemax.com/os/resources/learn/knowledgebase/simulation-of-young-s-interference-experiment-via>. [Accedido: Jun. 2018]
- [5] “Mach-Zehnder interferometer”, *EDU.Photonics.com*. [En línea]. Disponible en: https://www.photonics.com/d5274/Mach-Zehnder_interferometer. [Accedido: Jun. 2018]
- [6] L. Li, L. Xia, Z. Xie, y D. Liu, “All-fiber Mach-Zehnder interferometers for sensing applications”, *Opt. Express*, vol. 20, no. 10, pp. 11109–11120, May. 2012. doi: 10.1364/OE.20.011109.
- [7] L. Jiang, J. Yang, S. Wang, B. Li, y M. Wang, “Fiber Mach-Zehnder interferometer based on microcavities for high-temperature sensing with high sensitivity”, *Opt. Lett.*, vol. 36, no. 19, pp. 3753–3755, Oct. 2011. doi: 10.1364/OL.36.003753.
- [8] D. A. Jackson, J. E. Posada-Roman and J. A. Garcia-Souto, “New topologies for fiber laser non-contact vibrometers”, *IEEE SENSORS 2014 Proceedings*, Valencia, 2014, pp. 1531-1534. doi: 10.1109/ICSENS.2014.6985307.
- [9] A. Güemes, “Fiber Optics Strain Sensors”, *Struct. Health Monit. Mil. Veh.*, Berlín, 2014, pp. 67-93. doi: 10.14339/STO-EN-AVT-220.
- [10] “Vibración compuesta”, *Sinais ingeniería*. [En línea]. Disponible en: http://www.sinais.es/Recursos/Curso-vibraciones/fundamentos/vibracion_compuesta.html. [Accedido: Jun. 2018]
- [11] “Vibraciones en máquinas. Mantenimiento predictivo”, *IMAC: Ingeniería Mecánica Aplicada y Computacional*. [En línea]. Disponible en: http://www.imac.unavarra.es/web_imac/pages/docencia/asignaturas/emyv/pdfdoc/vib/vib_predictivo.pdf. [Accedido: Jun. 2018]
- [12] “Optical Heterodyne Detection”, *RP Photonics Encyclopedia*. [En línea]. Disponible en: https://www.rp-photonics.com/optical_heterodyne_detection.html. [Accedido: Jun. 2018]
- [13] D. Lin, J. Yan, Z. Chao, H. Jiang, y C. Yin, “Phasemeter with external trigger applied to PZT-modulated interferometer”, *Int. J. Electron.*, vol. 89, no. 10, pp. 759-769, 2002. doi: 10.1080/0020721031000093147.
- [14] P. Sandoz, T. Gharbi, y G. Tribillon, “Phase-shifting methods for interferometers using laser-diode frequency-modulation”, *Opt. Commun.*, vol. 132, no. 3, pp. 227-231, Dic. 1996. doi: 10.1016/0030-4018(96)00369-0.
- [15] S. Rerucha, Z. Buchta, M. Sarbort, J. Lazar, y O. Cip, “Detection of Interference Phase by Digital Computation of Quadrature Signals in Homodyne Laser Interferometry”, *Sensors*, vol. 12, no. 10, pp. 14095-14112, 2012. doi: 10.3390/s121014095.

- [16] L. Feng, J. He, J. Y. Duan, F. Li and Y. L. Liu, "Implementation of phase generated carrier technique for FBG Laser Sensor multiplexed system based on Compact RIO", *2008 1st Asia-Pacific Optical Fiber Sensors Conference*, Chengdu, 2008, pp. 1-4. doi: 10.1109/APOS.2008.5226295.
- [17] *PDB440-450 Series Operation Manual*, Thorlabs, Newton, NJ, 2014. [En línea]. Disponible en: <https://www.thorlabs.com/thorproduct.cfm?partnumber=PDB410A&pn=PDB410A>. [Accedido: Jun. 2018]
- [18] *UM1724 User manual*, STMicroelectronics, Plan-les-Ouates, Suiza, 2017. [En línea]. Disponible en: http://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf. [Accedido: Jun. 2018]
- [19] *RM0038 Reference manual*, STMicroelectronics, Plan-les-Ouates, Suiza, 2017. [En línea]. Disponible en: http://www.st.com/content/ccc/resource/technical/document/reference_manual/c/c/f9/93/b2/f0/82/42/57/CD00240193.pdf/files/CD00240193.pdf/jcr:content/translations/en.CD00240193.pdf. [Accedido: Jun. 2018]
- [20] M. Pronschinske, "The State of Native vs. Web vs. Hybrid", *DZone*, 16-jun-2014. [En línea]. Disponible en: <https://dzone.com/articles/state-native-vs-web-vs-hybrid>. [Accedido: Jun. 2018]
- [21] J.A. García-Souto, and H. Lamela, "Comparative analysis of optical-fibre interferometric sensors versus accelerometers: application to vibrations inside high-power transformers", *Journal of Optics A: Pure and Applied Optics*, vol. 4, no. 6, pp. 318-326, Nov. 2002.
- [22] *IEEE Guide for Induction Machinery Maintenance Testing and Failure Analysis*, IEEE Standard 1415, 2006. doi: 10.1109/IEEESTD.2007.357954.
- [23] J. A. Izatt, M. Choma, y C. Yang, "System and method for low coherence broadband quadrature interferometry", U.S. Patent US7019838B2, Mar. 28, 2006.
- [24] T. Terahara et al., "Quadrature phase-shift keying modulator and phase shift amount controlling method for the same", U.S. Patent US20090047028A1, Oct. 13, 2009.
- [25] "Machine learning for predictive maintenance: where to start?", *Medium*, 29-ago-2017. [En línea]. Disponible en: <https://medium.com/bigdatarepublic/machine-learning-for-predictive-maintenance-where-to-start-5f3b7586acfb>. [Accedido: Jun. 2018]

ANEXO I: ALGORITMO DE DETECCIÓN SÍNCRONA DE LA FASE ÓPTICA

ALGORITMO.M

```
%INTERFERÓMETRO

%Parámetros
t = linspace(0,40.96e-2,200000)'; %Tiempo y paso de simulación
I_0 = 2; %Intensidad proporcionada por el láser
I_max = 2; %Intensidad máxima detectada
I_min = 0; %Intensidad mínima detectada
P_0 = 0; %Fase óptica inicial
f_p = 20000; %Frecuencia de la señal de modulación
p_0 = pi/4; %Amplitud de la fase óptica de modulación
P_m = 5*sin(2*pi*100*t); %Fase óptica objeto de medida
lambda = 633*10^-9; %Longitud de onda del láser
n = 1.46; %Índice de refracción de la fibra óptica

%Simulación
w_p = 2*pi*f_p; %Frecuencia angular de la señal de modulación
P_p = P_0.*cos(w_p*t); %Fase óptica de modulación
V = (I_max-I_min)/(I_max+I_min); %Visibilidad del interferómetro
I = (1+cos(P_0+P_m+P_p).*V).*I_0/2; %Intensidad de salida del interferómetro

%Resultados
vin = cos(w_p*t)*100; %Tensión de entrada relativa (después de un filtro paso bajo)
vout = (I-I_0/2)/(I_0/2)*100; %Tensión de salida relativa (después de un filtro
paso bajo)

%-----

%ALGORITMO DE OBTENCIÓN DE LA FASE ÓPTICA MEDIANTE DETECCIÓN SÍNCRONA

%Variables utilizadas
vin_medio = (max(vin)+min(vin))/2; %Valor medio de vin para detectar los máximos y
mínimos de vin
tiempo1 = -1; %Índice del instante t1 en el que se pasa de vin<vin_medio a
vin>vin_medio
tiempo2 = -1; %Índice del instante t2 en el que se pasa de vin>vin_medio a
vin<vin_medio
maximos = []; %Índices de los máximos detectados en vin
minimos = []; %Índices de los mínimos detectados en vin
seno = []; %Señal detectada en los mínimos
coseno = []; %Señal detectada en los máximos (en cuadratura con la anterior)
muestras = -1; %Número de muestras de cada señal
fase = []; %Fase óptica instantánea
acumulada = 0; %Fase óptica acumulada
desplazamiento = []; %Desplazamiento equivalente
ts = []; %Instantes de muestreo
fs = -1; %Frecuencia de muestreo
muestras_fft = -1; %Número de muestras de la FFT
f = []; %Eje frecuencial de la FFT
fft_desplazamiento = []; %FFT de desplazamiento

%Extrae los máximos y mínimos de vin, y sus instantes de tiempo
for i=1:(size(vin)-1)
    if vin(i)<vin_medio && vin(i+1)>vin_medio
        tiempo1 = t(i);
    end
    if vin(i)>vin_medio && vin(i+1)<vin_medio && tiempo1~= -1
        tiempo2 = t(i);
        [M, I] = min(abs(t-(tiempo1+tiempo2)/2));
        maximos = [maximos; I];
    end
end
```



```

end
tiempo1 = -1;
tiempo2 = -1;
for i=1:(size(vin)-1)
    if vin(i)>vin_medio && vin(i+1)<vin_medio
        tiempo1 = t(i);
    end
    if vin(i)<vin_medio && vin(i+1)>vin_medio && tiempo1~= -1
        tiempo2 = t(i);
        [M, I] = min(abs(t-(tiempo1+tiempo2)/2));
        minimos = [minimos; I];
    end
end

%Separa las dos señales en cuadratura y se hace coincidir el número de muestras de cada una
seno = vout(minimos);
coseno = vout(maximos);
muestras = min(size(seno),size(coseno));
seno = seno(1:muestras);
coseno = coseno(1:muestras);

%Calcula la fase óptica instantánea de cada instante y la fase óptica total acumulada
fase = atan2(seno(1:muestras),coseno(1:muestras));
for i=1:(size(fase)-1)
    if fase(i)>0 && fase(i+1)<0 && abs(fase(i)-fase(i+1))>pi
        acumulada(i+1) = acumulada(i)+(2*pi-fase(i)+fase(i+1));
    elseif fase(i)<0 && fase(i+1)>0 && abs(fase(i)-fase(i+1))>pi
        acumulada(i+1) = acumulada(i)+(-2*pi-fase(i)+fase(i+1));
    else
        acumulada(i+1) = acumulada(i)+(fase(i+1)-fase(i));
    end
end

%Calcula el desplazamiento equivalente a partir de la fase óptica acumulada
desplazamiento = acumulada*lambda/(2*pi*n);

%Calcula la FFT del desplazamiento
ts = (t(maximos(1:muestras))+t(minimos(1:muestras)))/2;
fs = abs(1/(ts(1)-ts(2)));
muestras_fft = 2^nextpow2(length(desplazamiento));
f = (0:muestras_fft/2-1)*fs/muestras_fft;
fft_desplazamiento = fft(desplazamiento,muestras_fft)/muestras_fft*2;
fft_desplazamiento = abs(fft_desplazamiento(1:muestras_fft/2));

%-----
%REPRESENTACIÓN DE LOS RESULTADOS

figure;
plot(subplot(2,1,1),t*1000,vin);
hold on;
scatter(t(maximos)*1000,vin(maximos),[],'r');
scatter(t(minimos)*1000,vin(minimos),'b');
xlabel('t (ms)');
ylabel('V_i_r_e_l (%)');
xlim([0 10]);
title('a) Señal de modulación')
plot(subplot(2,1,2),t*1000,vin);
hold on;
scatter(t(maximos)*1000,vin(maximos),[],'r');
scatter(t(minimos)*1000,vin(minimos),'b');
xlabel('t (ms)');
ylabel('V_i_r_e_l (%)');
xlim([0 1]);
title('b) Detalle de la señal de modulación')

figure;
plot(subplot(2,1,1),t*1000,vout);
hold on;
scatter(t(minimos(1:muestras))*1000,seno,[],[1 0 0]);
scatter(t(maximos(1:muestras))*1000,coseno,'b');
xlabel('t (ms)');
ylabel('V_o_r_e_l (%)');
xlim([0 10]);
title('a) Señal de salida')

```

```

plot(subplot(2,1,2),t*1000,vout);
hold on;
scatter(t(minimos(1:muestras))*1000,seno,'r');
scatter(t(maximos(1:muestras))*1000,coseno,'b');
xlabel('t (ms)');
ylabel('V_o_r_e_l (%)');
xlim([0 2.5]);
title('b) Detalle de la señal de salida')

figure;
xlabel('t (ms)', 'FontWeight', 'bold');
yyaxis left
plot(ts*1000,fase);
plot(ts*1000,fase,'k:');
hold on;
plot(ts*1000,acumulada,'k');
ylabel('\Delta\phi (rad)', 'FontWeight', 'bold');
limites=ylim;
yyaxis right
ylim(limites*lambda/(2*pi*n)*10^9)
ylabel('\Deltad (nm)', 'FontWeight', 'bold');
xlim([0 25]);

figure;
xlabel('f (Hz)', 'FontWeight', 'bold');
yyaxis right
plot(f,fft_desplazamiento*10^9);
plot(f,fft_desplazamiento*10^9,'k');
ylabel('\Deltad (nm)', 'FontWeight', 'bold');
limites=ylim;
yyaxis left
ylim(limites/(lambda/(2*pi*n)*10^9))
ylabel('\Delta\phi (rad)', 'FontWeight', 'bold');
xlim([0 1000]);

```

ANEXO II: CÓDIGO FUENTE DEL MICROCONTROLADOR

MAIN.C

```
/**
*****
* @file      : main.c
* @brief     : Main program body
*****
** This notice applies to any and all portions of this file
* that are not between comment pairs USER CODE BEGIN and
* USER CODE END. Other portions of this file, whether
* inserted by the user or by software development tools
* are owned by their respective copyright owners.
*
* COPYRIGHT(c) 2018 STMicroelectronics
*
* Redistribution and use in source and binary forms, with or without modification,
* are permitted provided that the following conditions are met:
* 1. Redistributions of source code must retain the above copyright notice,
* this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright notice,
* this list of conditions and the following disclaimer in the documentation
* and/or other materials provided with the distribution.
* 3. Neither the name of STMicroelectronics nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
* OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*****
*/

/* Includes -----*/
#include "main.h"
#include "stm32l1xx_hal.h"
#include "Interferometro.h"

/* Private variables -----*/
ADC_HandleTypeDef hadc;
DAC_HandleTypeDef hdac;
DMA_HandleTypeDef hdma_dac_ch1;
TIM_HandleTypeDef htim2;
TIM_HandleTypeDef htim9;
UART_HandleTypeDef huart2;

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_DAC_Init(void);
static void MX_USART2_UART_Init(void);
static void MX_TIM9_Init(void);
static void MX_TIM2_Init(void);
static void MX_ADC_Init(void);
static void MX_NVIC_Init(void);

/**
* @brief The application entry point.
```

```

*
* @retval None
*/
int main(void)
{
    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* Configure the system clock */
    SystemClock_Config();

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_DAC_Init();
    MX_USART2_UART_Init();
    MX_TIM9_Init();
    MX_TIM2_Init();
    MX_ADC_Init();

    /* Initialize interrupts */
    MX_NVIC_Init();

    Inicializar_Interferometro();           //Inicializa las variables y configura los registros de
    los periféricos utilizados

    /* Infinite loop */
    while (1)
    {
        __WFI();                          //Entra en modo de reposo
    }
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct;
    RCC_ClkInitTypeDef RCC_ClkInitStruct;

    /**Configure the main internal regulator output voltage
    */
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /**Initializes the CPU, AHB and APB busses clocks
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSISate = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = 16;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL6;
    RCC_OscInitStruct.PLL.PLLDIV = RCC_PLL_DIV3;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    /**Initializes the CPU, AHB and APB busses clocks
    */
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                                |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSClkSource = RCC_SYSCCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
    {

```

```

    _Error_Handler(__FILE__, __LINE__);
}

/**Configure the Systick interrupt time
*/
HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

/**Configure the Systick
*/
HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

/* SysTick_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
}

/**
 * @brief NVIC Configuration.
 * @retval None
 */
static void MX_NVIC_Init(void)
{
    /* USART2_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(USART2_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(USART2_IRQn);
}

/* ADC init function */
static void MX_ADC_Init(void)
{
    ADC_ChannelConfTypeDef sConfig;

    /**Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of
    conversion)
    */
    hadc.Instance = ADC1;
    hadc.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV1;
    hadc.Init.Resolution = ADC_RESOLUTION_12B;
    hadc.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc.Init.ScanConvMode = ADC_SCAN_DISABLE;
    hadc.Init.EOCSelection = ADC_EOC_SEQ_CONV;
    hadc.Init.LowPowerAutoWait = ADC_AUTOWAIT_DISABLE;
    hadc.Init.LowPowerAutoPowerOff = ADC_AUTOPOWEROFF_DISABLE;
    hadc.Init.ChannelsBank = ADC_CHANNELS_BANK_A;
    hadc.Init.ContinuousConvMode = DISABLE;
    hadc.Init.NbrOfConversion = 1;
    hadc.Init.DiscontinuousConvMode = DISABLE;
    hadc.Init.ExternalTrigConv = ADC_EXTERNALTRIGCONV_T9_TRGO;
    hadc.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_RISING;
    hadc.Init.DMAContinuousRequests = DISABLE;
    if (HAL_ADC_Init(&hadc) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    /**Configure for the selected ADC regular channel its corresponding rank in the sequencer
    and its sample time.
    */
    sConfig.Channel = ADC_CHANNEL_0;
    sConfig.Rank = ADC_REGULAR_RANK_1;
    sConfig.SamplingTime = ADC_SAMPLETIME_4CYCLES;
    if (HAL_ADC_ConfigChannel(&hadc, &sConfig) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

/* DAC init function */
static void MX_DAC_Init(void)
{
    DAC_ChannelConfTypeDef sConfig;

```

```

    /**DAC Initialization
    */
    hdac.Instance = DAC;
    if (HAL_DAC_Init(&hdac) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    /**DAC channel OUT1 config
    */
    sConfig.DAC_Trigger = DAC_TRIGGER_T2_TRGO;
    sConfig.DAC_OutputBuffer = DAC_OUTPUTBUFFER_DISABLE;
    if (HAL_DAC_ConfigChannel(&hdac, &sConfig, DAC_CHANNEL_1) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

/* TIM2 init function */
static void MX_TIM2_Init(void)
{
    TIM_ClockConfigTypeDef sClockSourceConfig;
    TIM_MasterConfigTypeDef sMasterConfig;

    htim2.Instance = TIM2;
    htim2.Init.Prescaler = 0;
    htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim2.Init.Period = 31;
    htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    sMasterConfig.MasterOutputTrigger = TIM_TRGO_UPDATE;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_ENABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

/* TIM9 init function */
static void MX_TIM9_Init(void)
{
    TIM_SlaveConfigTypeDef sSlaveConfig;
    TIM_MasterConfigTypeDef sMasterConfig;

    htim9.Instance = TIM9;
    htim9.Init.Prescaler = 0;
    htim9.Init.CounterMode = TIM_COUNTERMODE_CENTERALIGNED1;
    htim9.Init.Period = 25;
    htim9.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    if (HAL_TIM_Base_Init(&htim9) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }

    sSlaveConfig.SlaveMode = TIM_SLAVEMODE_EXTERNAL1;
    sSlaveConfig.InputTrigger = TIM_TS_ITR0;
    if (HAL_TIM_SlaveConfigSynchronization(&htim9, &sSlaveConfig) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

```

```

sMasterConfig.MasterOutputTrigger = TIM_TRGO_UPDATE;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim9, &sMasterConfig) != HAL_OK)
{
    _Error_Handler(__FILE__, __LINE__);
}
}

/* USART2 init function */
static void MX_USART2_UART_Init(void)
{
    huart2.Instance = USART2;
    huart2.Init.BaudRate = 115200;
    huart2.Init.WordLength = UART_WORDLENGTH_8B;
    huart2.Init.StopBits = UART_STOPBITS_1;
    huart2.Init.Parity = UART_PARITY_NONE;
    huart2.Init.Mode = UART_MODE_TX_RX;
    huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart2.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart2) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}

/**
 * Enable DMA controller clock
 */
static void MX_DMA_Init(void)
{
    /* DMA controller clock enable */
    __HAL_RCC_DMA1_CLK_ENABLE();

    /* DMA interrupt init */
    /* DMA1_Channel2_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Channel2_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Channel2_IRQn);
}

/** Configure pins as
        * Analog
        * Input
        * Output
        * EVENT_OUT
        * EXTI
 */
static void MX_GPIO_Init(void)
{
    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOH_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
}

/**
 * @brief This function is executed in case of error occurrence.
 * @param file: The file name as string.
 * @param line: The line in file as a number.
 * @retval None
 */
void _Error_Handler(char *file, int line)
{
    /* User can add his own implementation to report the HAL error return state */
    while(1)
    {
    }
}

```

```
}
```

```
/***** (C) COPYRIGHT STMicroelectronics *****/
```

INTERFEROMETRO.H

```
/*
 * Interferometro.h
 */

#ifndef INTERFEROMETRO_H_
#define INTERFEROMETRO_H_

#define PI 3.141516
#define LAMBDA 633
#define FRECUENCIA 20000
#define MUESTRAS 2048
#define T_ESTABILIZACION 1000

#include "arm_math.h"
#include "math.h"
#include "stdlib.h"
#include "stdio.h"
#include "stm32l152xe.h"

void Inicializar_Interferometro(void);
void Delay_Ms(uint16_t milisegundos);
void EnviarCadena_USART(char* datos);
void EnviarUint16Array_USART(uint16_t* datos, uint16_t longitud);
void EnviarUint8Array_USART(uint8_t* datos, uint16_t longitud);
void EnviarInt8Array_USART(int8_t* datos, uint16_t longitud);
void EnviarInt16Array_USART(int16_t* datos, uint16_t longitud);
void EnviarFloat32Array_USART(float32_t* datos, uint16_t longitud);
void USART2_IRQHandler(void);

#endif /* INTERFEROMETRO_H_ */
```

INTERFEROMETRO.C

```
/*
 * Interferometro.c
 */

#include "Interferometro.h"

static const uint16_t vin[31][50] = { //Variables del sistema

{2110,2108,2106,2102,2098,2093,2088,2081,2074,2067,2060,2052,2044,2036,2029,2022,2015,2008,2003,1998,1994,1990,1988,1986,1986,1986,1988,1990,1994,1998,2003,2008,2015,2022,2029,2036,2044,2052,2060,2067,2074,2081,2088,2093,2098,2102,2106,2108,2110,2110},

{2171,2168,2163,2157,2148,2138,2127,2115,2101,2086,2071,2056,2040,2025,2010,1995,1981,1969,1958,1948,1939,1933,1928,1925,1924,1925,1928,1933,1939,1948,1958,1969,1981,1995,2010,2025,2040,2056,2071,2086,2101,2115,2127,2138,2148,2157,2163,2168,2171,2172},

{2233,2228,2221,2211,2199,2184,2167,2148,2127,2106,2083,2060,2036,2013,1990,1969,1948,1929,1912,1897,1885,1875,1868,1863,1862,1863,1868,1875,1885,1897,1912,1929,1948,1969,1990,2013,2036,2060,2083,2106,2127,2148,2167,2184,2199,2211,2221,2228,2233,2234},

{2294,2288,2279,2266,2249,2229,2206,2181,2154,2125,2095,2064,2032,2001,1971,1942,1915,1890,1867,1847,1830,1817,1808,1802,1800,1802,1808,1817,1830,1847,1867,1890,1915,1942,1971,2001,2032,2064,2095,2125,2154,2181,2206,2229,2249,2266,2279,2288,2294,2296},

{2356,2349,2337,2320,2299,2274,2246,2214,2180,2144,2106,2067,2029,1990,1952,1916,1882,1850,1822,1797,1776,1759,1747,1740,1738,1740,1747,1759,1776,1797,1822,1850,1882,1916,1952,1990,2029,2067,2106,2144,2180,2214,2246,2274,2299,2320,2337,2349,2356,2358},

{2417,2409,2394,2374,2349,2319,2285,2248,2207,2163,2118,2071,2025,1978,1933,1889,1848,1811,1777,1747,1722,1702,1687,1679,1676,1679,1687,1702,1722,1747,1777,1811,1848,1889,1933,1978,2025,2071,2118,2163,2207,2248,2285,2319,2349,2374,2394,2409,2417,2420},
```


{2479, 2469, 2452, 2429, 2399, 2365, 2325, 2281, 2233, 2182, 2129, 2075, 2021, 1967, 1914, 1863, 1815, 1771, 1731, 1697, 1667, 1644, 1627, 1617, 1614, 1617, 1627, 1644, 1667, 1697, 1731, 1771, 1815, 1863, 1914, 1967, 2021, 2075, 2129, 2182, 2233, 2281, 2325, 2365, 2399, 2429, 2452, 2469, 2479, 2482},

{2541, 2529, 2510, 2483, 2450, 2410, 2364, 2314, 2259, 2201, 2141, 2079, 2017, 1955, 1895, 1837, 1782, 1732, 1686, 1646, 1613, 1586, 1567, 1555, 1551, 1555, 1567, 1586, 1613, 1646, 1686, 1732, 1782, 1837, 1895, 1955, 2017, 2079, 2141, 2201, 2259, 2314, 2364, 2410, 2450, 2483, 2510, 2529, 2541, 2545},

{2602, 2589, 2567, 2537, 2500, 2455, 2404, 2347, 2286, 2221, 2153, 2083, 2013, 1943, 1875, 1810, 1749, 1692, 1641, 1596, 1559, 1529, 1507, 1494, 1489, 1494, 1507, 1529, 1559, 1596, 1641, 1692, 1749, 1810, 1875, 1943, 2013, 2083, 2153, 2221, 2286, 2347, 2404, 2455, 2500, 2537, 2567, 2589, 2602, 2607},

{2664, 2649, 2625, 2592, 2550, 2500, 2444, 2381, 2312, 2240, 2164, 2087, 2009, 1932, 1856, 1784, 1715, 1652, 1596, 1546, 1504, 1471, 1447, 1432, 1427, 1432, 1447, 1471, 1504, 1546, 1596, 1652, 1715, 1784, 1856, 1932, 2009, 2087, 2164, 2240, 2312, 2381, 2444, 2500, 2550, 2592, 2625, 2649, 2664, 2669},

{2725, 2709, 2683, 2646, 2600, 2546, 2483, 2414, 2339, 2259, 2176, 2091, 2005, 1920, 1837, 1757, 1682, 1613, 1550, 1496, 1450, 1413, 1387, 1371, 1365, 1371, 1387, 1413, 1450, 1496, 1550, 1613, 1682, 1757, 1837, 1920, 2005, 2091, 2176, 2259, 2339, 2414, 2483, 2546, 2600, 2646, 2683, 2709, 2725, 2731},

{2787, 2769, 2740, 2701, 2651, 2591, 2523, 2447, 2365, 2278, 2188, 2095, 2001, 1908, 1818, 1731, 1649, 1573, 1505, 1445, 1395, 1356, 1327, 1309, 1303, 1309, 1327, 1356, 1395, 1445, 1505, 1573, 1649, 1731, 1818, 1908, 2001, 2095, 2188, 2278, 2365, 2447, 2523, 2591, 2651, 2701, 2740, 2769, 2787, 2793},

{2848, 2829, 2798, 2755, 2701, 2636, 2562, 2480, 2392, 2297, 2199, 2099, 1997, 1897, 1799, 1704, 1616, 1534, 1460, 1395, 1341, 1298, 1267, 1248, 1241, 1248, 1267, 1298, 1341, 1395, 1460, 1534, 1616, 1704, 1799, 1897, 1997, 2099, 2199, 2297, 2392, 2480, 2562, 2636, 2701, 2755, 2798, 2829, 2848, 2855},

{2910, 2890, 2856, 2809, 2751, 2681, 2602, 2514, 2418, 2317, 2211, 2103, 1993, 1885, 1779, 1678, 1582, 1494, 1415, 1345, 1287, 1240, 1206, 1186, 1179, 1186, 1206, 1240, 1287, 1345, 1415, 1494, 1582, 1678, 1779, 1885, 1993, 2103, 2211, 2317, 2418, 2514, 2602, 2681, 2751, 2809, 2856, 2890, 2910, 2917},

{2972, 2950, 2914, 2864, 2801, 2727, 2641, 2547, 2444, 2336, 2222, 2106, 1990, 1874, 1760, 1652, 1549, 1455, 1369, 1295, 1232, 1182, 1146, 1124, 1117, 1124, 1146, 1182, 1232, 1295, 1369, 1455, 1549, 1652, 1760, 1874, 1990, 2106, 2222, 2336, 2444, 2547, 2641, 2727, 2801, 2864, 2914, 2950, 2972, 2979},

{3033, 3010, 2971, 2918, 2851, 2772, 2681, 2580, 2471, 2355, 2234, 2110, 1986, 1862, 1741, 1625, 1516, 1415, 1324, 1245, 1178, 1125, 1086, 1063, 1055, 1063, 1086, 1125, 1178, 1245, 1324, 1415, 1516, 1625, 1741, 1862, 1986, 2110, 2234, 2355, 2471, 2580, 2681, 2772, 2851, 2918, 2971, 3010, 3033, 3041},

{3095, 3070, 3029, 2973, 2902, 2817, 2721, 2613, 2497, 2374, 2246, 2114, 1982, 1850, 1722, 1599, 1483, 1375, 1279, 1194, 1123, 1067, 1026, 1001, 993, 1001, 1026, 1067, 1123, 1194, 1279, 1375, 1483, 1599, 1722, 1850, 1982, 2114, 2246, 2374, 2497, 2613, 2721, 2817, 2902, 2973, 3029, 3070, 3095, 3103},

{3156, 3130, 3087, 3027, 2952, 2862, 2760, 2647, 2524, 2393, 2257, 2118, 1978, 1839, 1703, 1572, 1449, 1336, 1234, 1144, 1069, 1009, 966, 940, 931, 940, 966, 1009, 1069, 1144, 1234, 1336, 1449, 1572, 1703, 1839, 1978, 2118, 2257, 2393, 2524, 2647, 2760, 2862, 2952, 3027, 3087, 3130, 3156, 3165},

{3218, 3190, 3144, 3081, 3002, 2908, 2800, 2680, 2550, 2412, 2269, 2122, 1974, 1827, 1684, 1546, 1416, 1296, 1188, 1094, 1015, 952, 906, 878, 869, 878, 906, 952, 1015, 1094, 1188, 1296, 1416, 1546, 1684, 1827, 1974, 2122, 2269, 2412, 2550, 2680, 2800, 2908, 3002, 3081, 3144, 3190, 3218, 3227},

{3280, 3250, 3202, 3136, 3052, 2953, 2839, 2713, 2577, 2432, 2281, 2126, 1970, 1815, 1664, 1519, 1383, 1257, 1143, 1044, 960, 894, 846, 816, 807, 816, 846, 894, 960, 1044, 1143, 1257, 1383, 1519, 1664, 1815, 1970, 2126, 2281, 2432, 2577, 2713, 2839, 2953, 3052, 3136, 3202, 3250, 3280, 3289},

{3341, 3310, 3260, 3190, 3102, 2998, 2879, 2746, 2603, 2451, 2292, 2130, 1966, 1804, 1645, 1493, 1350, 1217, 1098, 994, 906, 836, 786, 755, 745, 755, 786, 836, 906, 994, 1098, 1217, 1350, 1493, 1645, 1804, 1966, 2130, 2292, 2451, 2603, 2746, 2879, 2998, 3102, 3190, 3260, 3310, 3341, 3351},

{3403, 3371, 3318, 3245, 3153, 3043, 2918, 2780, 2629, 2470, 2304, 2134, 1962, 1792, 1626, 1467, 1316, 1178, 1053, 943, 851, 778, 725, 693, 683, 693, 725, 778, 851, 943, 1053, 1178, 1316, 1467, 1626, 1792, 1962, 2134, 2304, 2470, 2629, 2780, 2918, 3043, 3153, 3245, 3318, 3371, 3403, 3413},

{3464, 3431, 3375, 3299, 3203, 3089, 2958, 2813, 2656, 2489, 2315, 2138, 1958, 1781, 1607, 1440, 1283, 1138, 1007, 893, 797, 721, 665, 632, 621, 632, 665, 721, 797, 893, 1007, 1138, 1283, 1440, 1607, 1781, 1958, 2138, 2315, 2489, 2656, 2813, 2958, 3089, 3203, 3299, 3375, 3431, 3464, 3475},

{3526, 3491, 3433, 3353, 3253, 3134, 2997, 2846, 2682, 2508, 2327, 2142, 1954, 1769, 1588, 1414, 1250, 1099, 962, 843, 743, 663, 605, 570, 558, 570, 605, 663, 743, 843, 962, 1099, 1250, 1414, 1588, 1769, 1954, 2142, 2327, 2508, 2682, 2846, 2997, 3134, 3253, 3353, 3433, 3491, 3526, 3538},

```
{3587, 3551, 3491, 3408, 3303, 3179, 3037, 2879, 2709, 2527, 2339, 2145, 1951, 1757, 1569, 1387, 1217, 1059, 917, 793, 688, 605, 545, 509, 496, 509, 545, 605, 688, 793, 917, 1059, 1217, 1387, 1569, 1757, 1951, 2145, 2339, 2527, 2709, 2879, 3037, 3179, 3303, 3408, 3491, 3551, 3587, 3600},
```

```
{3649, 3611, 3548, 3462, 3353, 3224, 3077, 2913, 2735, 2547, 2350, 2149, 1947, 1746, 1549, 1361, 1183, 1019, 872, 743, 634, 548, 485, 447, 434, 447, 485, 548, 634, 743, 872, 1019, 1183, 1361, 1549, 1746, 1947, 2149, 2350, 2547, 2735, 2913, 3077, 3224, 3353, 3462, 3548, 3611, 3649, 3662},
```

```
{3711, 3671, 3606, 3516, 3404, 3270, 3116, 2946, 2761, 2566, 2362, 2153, 1943, 1734, 1530, 1335, 1150, 980, 826, 692, 580, 490, 425, 385, 372, 385, 425, 490, 580, 692, 826, 980, 1150, 1335, 1530, 1734, 1943, 2153, 2362, 2566, 2761, 2946, 3116, 3270, 3404, 3516, 3606, 3671, 3711, 3724},
```

```
{3772, 3731, 3664, 3571, 3454, 3315, 3156, 2979, 2788, 2585, 2374, 2157, 1939, 1722, 1511, 1308, 1117, 940, 781, 642, 525, 432, 365, 324, 310, 324, 365, 432, 525, 642, 781, 940, 1117, 1308, 1511, 1722, 1939, 2157, 2374, 2585, 2788, 2979, 3156, 3315, 3454, 3571, 3664, 3731, 3772, 3786},
```

```
{3834, 3791, 3721, 3625, 3504, 3360, 3195, 3012, 2814, 2604, 2385, 2161, 1935, 1711, 1492, 1282, 1084, 901, 736, 592, 471, 375, 305, 262, 248, 262, 305, 375, 471, 592, 736, 901, 1084, 1282, 1492, 1711, 1935, 2161, 2385, 2604, 2814, 3012, 3195, 3360, 3504, 3625, 3721, 3791, 3834, 3848},
```

```
{3895, 3851, 3779, 3680, 3554, 3405, 3235, 3046, 2841, 2623, 2397, 2165, 1931, 1699, 1473, 1255, 1050, 861, 691, 542, 416, 317, 245, 201, 186, 201, 245, 317, 416, 542, 691, 861, 1050, 1255, 1473, 1699, 1931, 2165, 2397, 2623, 2841, 3046, 3235, 3405, 3554, 3680, 3779, 3851, 3895, 3910},
```

```
{3957, 3912, 3837, 3734, 3605, 3451, 3274, 3079, 2867, 2643, 2409, 2169, 1927, 1687, 1453, 1229, 1017, 822, 645, 491, 362, 259, 184, 139, 124, 139, 184, 259, 362, 491, 645, 822, 1017, 1229, 1453, 1687, 1927, 2169, 2409, 2643, 2867, 3079, 3274, 3451, 3605, 3734, 3837, 3912, 3957, 3972}
```

```
};
```

```
static uint16_t lambda;
static uint16_t frecuencia;
static uint16_t muestras;
static uint8_t amplitud;
static uint16_t vout[MUESTRAS];
static int16_t seno[MUESTRAS/2];
static int16_t coseno[MUESTRAS/2];
static float32_t fase[MUESTRAS/2];
static float32_t acumulada[MUESTRAS/2];
static float32_t desplazamiento[MUESTRAS/2];
static float32_t fft1[MUESTRAS/2];
static float32_t fft2[MUESTRAS/2];
static float32_t fft3[MUESTRAS/2];
static arm_rfft_fast_instance_f32 ffti;
```

```
void Inicializar_Interferometro(void) //Inicializa y configura los periféricos necesarios
```

```
{
    lambda = LAMBDA;
    frecuencia = FRECUENCIA;
    muestras = MUESTRAS;
    amplitud = 15;

    ADC1->CR1 |= ADC_CR1_EOCIE;
    DAC->CR |= DAC_CR_DMAEN1;
    DAC->CR |= DAC_CR_DMAUDRIE1;
    DAC->CR |= DAC_CR_EN1;
    DMA1->IFCR |= DMA_IFCR_CGIF2;
    DMA1_Channel2->CNDTR = 50;
    DMA1_Channel2->CMAR = (uint32_t)vin[amplitud];
    DMA1_Channel2->CPAR = (uint32_t)&DAC->DHR12R1;
    DMA1_Channel2->CCR |= DMA_CCR_EN;
    TIM2->ARR = 640000 / FRECUENCIA - 1;
    TIM9->CR1 |= TIM_CR1_CEN;
    TIM2->CR1 |= TIM_CR1_CEN;
    USART2->CR1 |= USART_CR1_RXNEIE;
}
```

```
void Delay_Ms(uint16_t milisegundos) //Genera un retraso de milisegundos
```

```
{
    RCC->APB1ENR |= RCC_APB1ENR_TIM6EN;
    TIM6->CR1 &= ~TIM_CR1_CEN;
    TIM6->CNT = 0;
    TIM6->PSC = 32000;
    TIM6->ARR = milisegundos;
```

```

    TIM6->SR &= ~TIM_SR_UIF;
    TIM6->CR1 |= TIM_CR1_CEN;
    while(!TIM6->SR & TIM_SR_UIF) {}
    TIM6->SR &= ~TIM_SR_UIF;
    TIM6->CR1 &= ~TIM_CR1_CEN;
}

void EnviarCadena_USART(char* datos) //Envía una cadena de caracteres por el puerto
serie mediante USART2
{
    int longitud = -1;
    int i = 0;

    while(longitud == -1)
    {
        if(datos[i] == '\0')
        {
            longitud = i;
        }
        else
        {
            i++;
        }
    }
    for(i = 0; i < longitud; i++)
    {
        USART2->DR = datos[i];
        while(!(USART2->SR & USART_SR_TXE)) {}
    }
}

void EnviarUint16Array_USART(uint16_t* datos, uint16_t longitud) //Envía un array de
uint16 por el puerto serie mediante USART2
{
    char cadena[6]; //Caracteres máximos: 5 (65535) + 1 ('\0')

    EnviarCadena_USART("{");
    for(int i = 0; i < longitud; i++)
    {
        sprintf(cadena, "%d", datos[i]);
        EnviarCadena_USART(cadena);
        if(i < longitud - 1)
        {
            EnviarCadena_USART(",");
        }
    }
    EnviarCadena_USART("}");
}

void EnviarUint8Array_USART(uint8_t* datos, uint16_t longitud) //Envía un array de
uint8 por el puerto serie mediante USART2
{
    char cadena[4]; //Caracteres máximos: 3 (255) + 1 ('\0')

    EnviarCadena_USART("{");
    for(int i = 0; i < longitud; i++)
    {
        sprintf(cadena, "%d", datos[i]);
        EnviarCadena_USART(cadena);
        if(i < longitud - 1)
        {
            EnviarCadena_USART(",");
        }
    }
    EnviarCadena_USART("}");
}

void EnviarInt8Array_USART(int8_t* datos, uint16_t longitud) //Envía un array de int8
por el puerto serie mediante USART2
{
    char cadena[5]; //Caracteres máximos: 1 (signo) + 3 (128) + 1 ('\0')

    EnviarCadena_USART("{");
    for(int i = 0; i < longitud; i++)

```

```

    {
        sprintf(cadena, "%d", datos[i]);
        EnviarCadena_USART(cadena);
        if(i < longitud - 1)
        {
            EnviarCadena_USART(",");
        }
    }
    EnviarCadena_USART("{}");
}

void EnviarInt16Array_USART(int16_t* datos, uint16_t longitud) //Envía un array de
int16 por el puerto serie mediante USART2
{
    char cadena[7]; //Caracteres máximos: 1 (signo) + 5 (32768) + 1 ('\0')

    EnviarCadena_USART("{}");
    for(int i = 0; i < longitud; i++)
    {
        sprintf(cadena, "%d", datos[i]);
        EnviarCadena_USART(cadena);
        if(i < longitud - 1)
        {
            EnviarCadena_USART(",");
        }
    }
    EnviarCadena_USART("{}");
}

void EnviarFloat32Array_USART(float32_t* datos, uint16_t longitud) //Envía un array
de float32 por el puerto serie mediante USART2
{
    char cadena[14]; //Caracteres máximos (precisión decidida): 1 (signo) + 5 (parte entera)
+ 1 (punto) + 6 (parte decimal) + 1 ('\0') [12 dedicados al número completo]

    EnviarCadena_USART("{}");
    for(int i = 0; i < longitud; i++)
    {
        sprintf(cadena, "%f", datos[i]);
        EnviarCadena_USART(cadena);
        if(i < longitud - 1)
        {
            EnviarCadena_USART(",");
        }
    }
    cadena[13]='\0';
    EnviarCadena_USART("{}");
}

void USART2_IRQHandler(void) //RAI de USART2 (interrupción por recepción asíncrona)
{
    USART2->CR1 &= ~USART_CR1_RXNEIE;
    switch(USART2->DR)
    {
        case '1': //Modifica la lambda del láser
        {
            uint16_t lambda1;

            while(!(USART2->SR & USART_SR_RXNE)) {}
            *((uint8_t*)&lambda1)=USART2->DR;
            while(!(USART2->SR & USART_SR_RXNE)) {}
            *((uint8_t*)&lambda1)+1=USART2->DR;

            if(lambda1 > 0)
            {
                lambda = lambda1;
                EnviarCadena_USART("iLambda modificada\r\n");
            }
            else
            {
                EnviarCadena_USART("eLambda no modificada\r\n");
            }
            break;
        }
    }
}

```

```

case 'f':          //Modifica la frecuencia de modulación
{
    uint16_t frecuencia1;

    while(!(USART2->SR & USART_SR_RXNE)) {}
    *((uint8_t*)&frecuencia1)=USART2->DR;
    while(!(USART2->SR & USART_SR_RXNE)) {}
    *((uint8_t*)&frecuencia1)+1=USART2->DR;

    if(frecuencia1 > 0 && frecuencia1 <= FRECUENCIA)
    {
        frecuencia = frecuencia1;
        TIM2->ARR = 640000 / frecuencia - 1;
        EnviarCadena_USART("iFrecuencia modificada\r\n");
    }
    else
    {
        EnviarCadena_USART("eFrecuencia no modificada\r\n");
    }
    break;
}
case 'n':          //Modifica el número de muestras
{
    uint16_t muestras1;

    while(!(USART2->SR & USART_SR_RXNE)) {}
    *((uint8_t*)&muestras1)=USART2->DR;
    while(!(USART2->SR & USART_SR_RXNE)) {}
    *((uint8_t*)&muestras1)+1=USART2->DR;

    if((muestras1 & (muestras1 - 1)) == 0 && muestras1 <= MUESTRAS)
    {
        muestras = muestras1;
        EnviarCadena_USART("iNumero de muestras modificado\r\n");
    }
    else
    {
        EnviarCadena_USART("eNumero de muestras no modificado\r\n");
    }
    break;
}
case 'v':          //Modifica la amplitud de la señal de modulación
{
    uint8_t sentido;

    while(!(USART2->SR & USART_SR_RXNE)) {}
    sentido = USART2->DR;

    switch(sentido)
    {
        case 'i':          //Incrementa
        {
            if(amplitud < 30)
            {
                amplitud++;
                TIM2->CR1 &= ~TIM_CR1_CEN;
                TIM9->CR1 &= ~TIM_CR1_CEN;
                TIM2->CNT = 0;
                TIM9->CNT = 0;
                DMA1_Channel12->CCR &= ~DMA_CCR_EN;
                DMA1_Channel12->CNDTR = 50;
                DMA1_Channel12->CMAR = (uint32_t)vin[amplitud];
                DMA1_Channel12->CCR |= DMA_CCR_EN;
                TIM9->CR1 |= TIM_CR1_CEN;
                TIM2->CR1 |= TIM_CR1_CEN;
                EnviarCadena_USART("v");
                EnviarUint8Array_USART(&amplitud,1);
                EnviarCadena_USART("\r\n");
            }
            else
            {
                EnviarCadena_USART("eAmplitud maxima\r\n");
            }
            break;
        }
    }
}

```

```

    }
    case 'd': //Decrementa
    {
        if(amplitud > 0)
        {
            amplitud--;
            TIM2->CR1 &= ~TIM_CR1_CEN;
            TIM9->CR1 &= ~TIM_CR1_CEN;
            TIM2->CNT = 0;
            TIM9->CNT = 0;
            DMA1_Channel12->CCR &= ~DMA_CCR_EN;
            DMA1_Channel12->CNDTR = 50;
            DMA1_Channel12->CMAR = (uint32_t)vin[amplitud];
            DMA1_Channel12->CCR |= DMA_CCR_EN;
            TIM9->CR1 |= TIM_CR1_CEN;
            TIM2->CR1 |= TIM_CR1_CEN;
            EnviarCadena_USART("v");
            EnviarUint8Array_USART(&amplitud,1);
            EnviarCadena_USART("\r\n");
        }
        else
        {
            EnviarCadena_USART("eAmplitud minima\r\n");
        }
        break;
    }
    default:
    {
        EnviarCadena_USART("eAmplitud no modificada\r\n");
    }
}
break;
}
case 'r': //Devuelve el rango actual de adquisición en porcentajes
{
    uint16_t minimo_maximo[2] = {4095,0};
    uint16_t trama[25];

    ADC1->CR2 &= ~ADC_CR2_EXTSEL; //Configura el TIM2 como trigger
    del ADC para medir el rango de adquisición del ADC
    ADC1->CR2 |= ADC_CR2_EXTSEL_2 | ADC_CR2_EXTSEL_1;
    while(DMA1_Channel12->CNDTR != 25) {} //Espera al
    siguiente mínimo de la señal de modulación
    ADC1->SR &= ~ADC_SR_EOC; //Enciende el ADC
    ADC1->SR &= ~ADC_SR_OVR;
    ADC1->CR2 |= ADC_CR2_ADON;
    for(uint8_t i = 0; i < 25; i++) //Adquiere muestras durante un
    semiperiodo de la señal de modulación
    {
        while(!(ADC1->SR & ADC_SR_EOC)) {}
        trama[i]=ADC1->DR;
    }
    ADC1->CR2 &= ~ADC_CR2_ADON; //Apaga el ADC
    ADC1->CR2 &= ~ADC_CR2_EXTSEL; //Configura el TIM9 como trigger
    del ADC para volver a realizar mediciones
    ADC1->CR2 |= ADC_CR2_EXTSEL_0;

    for(uint8_t i = 0; i < 25; i++) //Extrae el máximo y el mínimo
    detectados
    {
        if (trama[i] < minimo_maximo[0])
        {
            minimo_maximo[0] = trama[i];
        }
        if (trama[i] > minimo_maximo[1])
        {
            minimo_maximo[1] = trama[i];
        }
    }
    minimo_maximo[0] = (minimo_maximo[0] * 100) / 4095;
    //Calcula los porcentajes correspondientes
    minimo_maximo[1] = (minimo_maximo[1] * 100) / 4095;

    EnviarCadena_USART("r");
}

```

```

        EnviarUint16Array_USART(minimo_maximo,2);
        EnviarCadena_USART("\r\n");
        break;
    }
    case 'a':          //Rutina de autocalibración del sistema
    {
        uint8_t calibrado = 0;
        int8_t resultado[2];
        uint16_t trama[25];
        float32_t posiciones[3] = {-1,-1,-1};
        float32_t referencia = 130989.9/(float32_t)frecuencia-0.1;
        float32_t diferencia = 2*referencia;
        float32_t diferencia1;

        ADC1->CR2 &= ~ADC_CR2_EXTSEL;          //Configura el TIM2 como trigger
        ADC1->CR2 |= ADC_CR2_EXTSEL_2 | ADC_CR2_EXTSEL_1;
        amplitud = 0xFF;

        do
        {
            amplitud++;          //Cambia la amplitud de la señal de
            modulación

            TIM2->CR1 &= ~TIM_CR1_CEN;
            TIM9->CR1 &= ~TIM_CR1_CEN;
            TIM2->CNT = 0;
            TIM9->CNT = 0;
            DMA1_Channel12->CCR &= ~DMA_CCR_EN;
            DMA1_Channel12->CNDTR = 50;
            DMA1_Channel12->CMAR = (uint32_t)vin[amplitud];
            DMA1_Channel12->CCR |= DMA_CCR_EN;
            TIM9->CR1 |= TIM_CR1_CEN;
            TIM2->CR1 |= TIM_CR1_CEN;

            Delay_Ms(T_ESTABILIZACION);          //Espera a estabilizarse
            el sistema antes de calibrar

            while(DMA1_Channel12->CNDTR != 25) {}          //Espera
            al siguiente mínimo de la señal de modulación
            ADC1->SR &= ~ADC_SR_EOC;          //Enciende el ADC
            ADC1->SR &= ~ADC_SR_OVR;
            ADC1->CR2 |= ADC_CR2_ADON;
            for(uint8_t i = 0; i < 25; i++)          //Adquiere muestras
            durante un semiperiodo de la señal de modulación
            {
                while(!(ADC1->SR & ADC_SR_EOC)) {}
                trama[i]=ADC1->DR;
            }
            ADC1->CR2 &= ~ADC_CR2_ADON;          //Apaga el ADC

            for(uint8_t i = 0; i < 24; i++)          //Detecta los cruces por
            cero
            {
                if(((trama[i] < 2048) && (trama[i+1] > 2048)) ||
                ((trama[i] > 2048) && (trama[i+1] < 2048)))
                {
                    if(posiciones[0] == -1)
                    {
                        posiciones[0] = i;
                    }
                    else if(posiciones[1] == -1)
                    {
                        posiciones[1] = i;
                    }
                    else
                    {
                        posiciones[2] = i;
                    }
                }
            }
            if(posiciones[0] != -1 && posiciones[1] != -1)
            //Comprueba que existen al menos dos cruces por cero
            {

```

```

    posiciones[0]=(1/((float32_t)trama[(uint8_t)posiciones[0]+1]-
(float32_t)trama[(uint8_t)posiciones[0]]))*(2048-
(float32_t)trama[(uint8_t)posiciones[0]])+(float32_t)posiciones[0];

    posiciones[1]=(1/((float32_t)trama[(uint8_t)posiciones[1]+1]-
(float32_t)trama[(uint8_t)posiciones[1]]))*(2048-
(float32_t)trama[(uint8_t)posiciones[1]])+(float32_t)posiciones[1];
    diferencia = posiciones[1] - posiciones[0];
    if(posiciones[2] != -1) //Analiza el
tercer cruce si existe
    {
        posiciones[2] =
(1/((float32_t)trama[(uint8_t)posiciones[2]+1]-
(float32_t)trama[(uint8_t)posiciones[2]]))*(2048-
(float32_t)trama[(uint8_t)posiciones[2]])+(float32_t)posiciones[2];
        diferencial = posiciones[2] - posiciones[1];
        if(diferencial < diferencia)
        {
            diferencia = diferencial;
        }
    }
    if(diferencia < referencia) //Comprueba la
diferencia de tiempo
    {
        calibrado = 1;
    }
    posiciones[0] = -1;
    posiciones[1] = -1;
    posiciones[2] = -1;
}
while(amplitud < 30 && calibrado == 0);

ADC1->CR2 &= ~ADC_CR2_EXTSEL; //Configura el TIM9 como trigger
del ADC para volver a realizar mediciones
ADC1->CR2 |= ADC_CR2_EXTSEL_0;

if(calibrado == 1)
{
    EnviarCadena_USART("a");
    resultado[0]=amplitud;
    resultado[1]=(int8_t)((((diferencia-referencia)*100)/referencia);
    EnviarInt8Array_USART(resultado,2);
    EnviarCadena_USART("\r\n");
}
else
{
    EnviarCadena_USART("a");
    resultado[0]=-1;
    resultado[1]=(int8_t)((((diferencia-referencia)*100)/referencia);
    EnviarInt8Array_USART(resultado,2);
    EnviarCadena_USART("\r\n");
}
break;
}
case 'm': //Realiza una medición
{
    while(DMA1_Channel2->CNDTR != 25) {} //Espera al siguiente
mínimo de la señal de modulación
    ADC1->SR &= ~ADC_SR_EOC; //Enciende el ADC
    ADC1->SR &= ~ADC_SR_OVR;
    ADC1->CR2 |= ADC_CR2_ADON;
    for(uint16_t i = 0; i < muestras; i++) //Adquiere el número de
muestras definido
    {
        while(!(ADC1->SR & ADC_SR_EOC)) {}
        vout[i]=ADC1->DR;
    }
    ADC1->CR2 &= ~ADC_CR2_ADON; //Apaga el ADC

    for(uint16_t i = 0; i < muestras/2; i++) //Calcula el seno
    {
        seno[i]=vout[i*2]-2048;
    }
}

```



```

for(uint16_t i = 0; i < muestras/2; i++)           //Calcula el coseno
{
    coseno[i]=vout[1+i*2]-2048;
}
for(uint16_t i = 0; i < muestras/2; i++)           //Calcula la fase óptica
{
    fase[i]=atan((float32_t)seno[i]/(float32_t)coseno[i]);
}
acumulada[0]=0;
for(uint16_t i = 0; i < muestras/2-1; i++)           //Calcula la
fase óptica acumulada
{
    if(fase[i] > 0 && fase[i+1] < 0 && abs(fase[i] - fase[i+1]) >
PI/2)
    {
        acumulada[i+1]=acumulada[i]+(PI-fase[i]+fase[i+1]);
    }
    else if(fase[i] < 0 && fase[i+1] > 0 && abs(fase[i] - fase[i+1])
> PI/2)
    {
        acumulada[i+1]=acumulada[i]+(-PI-fase[i]+fase[i+1]);
    }
    else
    {
        acumulada[i+1]=acumulada[i]+(fase[i+1]-fase[i]);
    }
}
for(uint16_t i = 0; i < muestras/2; i++)           //Calcula desplazamiento
(t[] = (0:((MUESTRAS/2)/2-1))*1/FRECUENCIA [s])
{
    desplazamiento[i]=acumulada[i]*lambda/(2*PI);
}
for(uint16_t i = 0; i < muestras/2; i++)           //Calcula la FFT de
desplazamiento (f[] = (0:((MUESTRAS/2)/2-1))*FRECUENCIA/(MUESTRAS/2) [Hz])
{
    fft1[i]=desplazamiento[i];
}
arm_rfft_fast_init_f32(&ffti,muestras/2);
arm_rfft_fast_f32(&ffti,fft1,fft2,0);
arm_cmplx_mag_f32(fft2,fft3,muestras/2);
for(uint16_t i = 0; i < muestras/4; i++)
{
    fft3[i]=fft3[i]/(muestras/2)*2;
}

EnviarCadena_USART("m");
EnviarUint16Array_USART(vout,muestras);
EnviarCadena_USART("\r\n");
break;
}
case 'g':           //Devuelve los datos adquiridos y procesados
{
    uint8_t grafico;
    while(!(USART2->SR & USART_SR_RXNE)) {}
    grafico = USART2->DR;

    switch(grafico)
    {
        case 'm':           //Muestras
        {
            EnviarCadena_USART("m");
            EnviarUint16Array_USART(vout,muestras);
            EnviarCadena_USART("\r\n");
            break;
        }
        case 's':           //Seno
        {
            EnviarCadena_USART("s");
            EnviarInt16Array_USART(seno,muestras/2);
            EnviarCadena_USART("\r\n");
            break;
        }
        case 'c':           //Coseno

```

```

        {
            EnviarCadena_USART("c");
            EnviarInt16Array_USART(coseno,muestras/2);
            EnviarCadena_USART("\r\n");
            break;
        }
        case 'p':                //Fase
        {
            EnviarCadena_USART("p");
            EnviarFloat32Array_USART(acumulada,muestras/2);
            EnviarCadena_USART("\r\n");
            break;
        }
        case 'd':                //desplazamiento
        {
            EnviarCadena_USART("d");
            EnviarFloat32Array_USART(desplazamiento,muestras/2);
            EnviarCadena_USART("\r\n");
            break;
        }
        case 'f':                //FFT
        {
            EnviarCadena_USART("f");
            EnviarFloat32Array_USART(fft3,muestras/4);
            EnviarCadena_USART("\r\n");
            break;
        }
        default:
        {
            EnviarCadena_USART("eDatos no reconocidos\r\n");
        }
    }
    break;
}
case 'i':                //Inicializa las variables a sus valores predeterminados
{
    Inicializar_Interferometro();

    EnviarCadena_USART("iParametros inicializados\r\n");
    break;
}
default:
{
    EnviarCadena_USART("eComando incorrecto\r\n");
}
}
USART2->CR1 |= USART_CR1_RXNEIE;
}

```

ANEXO III: PROTOCOLO DE COMUNICACIÓN MEDIANTE USART

La configuración de la comunicación USART entre el microcontrolador y el otro dispositivo (en este caso un ordenador) es la siguiente:

- **Baud Rate:** 115200 bps
- **Word Length:** 8 bits
- **Parity:** ninguna
- **Mode:** transmisión (TX) y recepción (RX)
- **Hardware Flow Control:** ninguno
- **Oversampling:** x16

A continuación, se explica el protocolo de comunicación que debe seguirse para comunicarse con el sistema. El formato de los datos que se debe adoptar para enviar datos al microcontrolador se muestra en la **Tabla 6**.

Tabla 6. Formato de los datos enviados al microcontrolador

DATO ENVIADO AL MICROCONTROLADOR	
Comando	(Parámetro)
1 byte	1 o 2 bytes

Un dato enviado al microcontrolador consta de un comando representado por un carácter en codificación ASCII y un parámetro (si es necesario) que puede ser un carácter (codificación ASCII) o un número entero positivo de 8 o 16 bits.

El formato de los datos recibidos del microcontrolador se muestra en la **Tabla 7**.

Tabla 7. Formato de los datos recibidos del microcontrolador

DATO RECIBIDO DEL MICROCONTROLADOR		
Tipo de respuesta	Contenido	Final de línea
1 byte	1 o más bytes	\r\n (2 bytes)

El “Final de línea” indica el final de la respuesta, puesto que pueden recibirse cadenas largas de caracteres o vectores de números correspondientes a una medición realizada. El formato de

“Contenido” es ASCII, y se debe tratar siempre como si fuera una cadena de caracteres. Los vectores de números (enviados como una cadena de caracteres) comienzan con ‘{’, contienen los números separados con ‘,’ y terminan con ‘}’. El ‘.’ separa la parte entera y decimal de un número de tipo flotante. El campo “Tipo de respuesta” puede tener distintos valores (**Tabla 8**), y es utilizado por la aplicación para distinguir la información recibida y utilizarla según corresponda.

Tabla 8. Tipos de respuesta

TIPOS DE RESPUESTA	
TIPO	DESCRIPCIÓN
‘i’	Cadena de caracteres que contiene un mensaje informativo
‘e’	Cadena de caracteres que informa de un error en las acciones realizadas
Otro	El carácter es el mismo que el del comando enviado. Se devuelve un array de números.

A continuación, la **Tabla 9** resume las acciones de cada uno de los comandos, así como las posibles respuestas de cada uno.

Tabla 9. Comandos del programa del microcontrolador

COMANDO	DESCRIPCIÓN	RESPUESTA/S
“l[LAMBDA]”	Modifica la longitud de onda del láser. Debe indicarse la misma dividida por el índice de refracción de la fibra óptica. <i>LAMBDA</i> = Número entero positivo mayor que cero (16 bits). Unidades en nm.	<i>“iLambda modificada\r\n”</i> o <i>“eLambda no modificada\r\n”</i>
“f[FRECUENCIA]”	Modifica la frecuencia de modulación. <i>FRECUENCIA</i> = Número entero positivo mayor que cero y menor o igual que 20000 (16 bits). Unidades en Hz.	<i>“iFrecuencia modificada\r\n”</i> o <i>“eFrecuencia no modificada\r\n”</i>
“n[NÚMERO DE MUESTRAS]”	Modifica el número de muestras a adquirir considerando los máximos y los mínimos. <i>NÚMERO DE MUESTRAS</i> = Potencia de 2, mayor o igual que 2 y menor o igual que 2048	<i>“iNumero de muestras modificado\r\n”</i> o <i>“eNumero de muestras no modificado\r\n”</i>

COMANDO	DESCRIPCIÓN	RESPUESTA/S
"v[SENTIDO]"	<p>Modifica la amplitud de la tensión de la señal de modulación en el pin de salida.</p> <p>SENTIDO = 'i' (incrementa) o 'd' (decrementa)</p>	<p><i>"v{[AMPLITUD]}\r\n", "eAmplitud maxima\r\n", "eAmplitud minima\r\n" o "eAmplitud no modificada\r\n"</i></p> <p>AMPLITUD = Número entre 0 y 30 (orden ascendente de amplitudes)</p>
"r"	Devuelve el rango actual de adquisición.	<p><i>"r{[MINIMO],[MAXIMO]}\r\n"</i></p> <p>MINIMO/ MAXIMO = Mínimo/ Máximo detectados en el ADC en porcentaje respecto a su rango máximo</p>
"a"	Ejecuta la rutina de calibración automática del sistema.	<p><i>"a{[AMPLITUD],[ERROR EN PORCENTAJE]}\r\n"</i></p> <p>AMPLITUD = Número de la amplitud (entre 0 y 30) o -1 (sistema no calibrado)</p> <p>ERROR EN PORCENTAJE = Error en la calibración en porcentaje (negativo si se ha sobremodulado)</p>
"m"	Realiza una medición.	<p><i>"m{[MEDIDAS]}\r\n"</i></p> <p>MEDIDAS = Vector de números enteros positivos (de 0 a 4095) adquirido en el ADC</p>
"g[DATOS]"	<p>Devuelve los datos adquiridos y procesados.</p> <p>DATOS = 'm' (muestras, en unidades del ADC sin signo), 's' (seno, en unidades del ADC con signo), 'c' (coseno, en unidades del ADC con signo), 'p' (fase, en radianes), 'd' (desplazamiento, en nm) o 'f' (FFT*, en nm)</p> <p><i>*El eje de frecuencias de la FFT debe construirse con la frecuencia de modulación y el número de muestras</i></p>	<p><i>"[PARÁMETRO]{[CONTENIDO]}\r\n" o "eDatos no reconocidos\r\n"</i></p> <p>PARÁMETRO = parámetro enviado (DATOS)</p> <p>CONTENIDO = vector de números de los datos solicitados</p>

COMANDO	DESCRIPCIÓN	RESPUESTA/S
"i"	Inicializa las variables de control a sus valores predeterminados.	<i>"iParametros inicializados\r\n"</i>
<i>Desconocido</i>	Devuelve un mensaje de error.	<i>"eComando incorrecto\r\n"</i>

ANEXO IV: CÓDIGO FUENTE DE LA APLICACIÓN INFORMÁTICA

APP.COMPONENT.HTML

```
<div style="text-align:center">

    <mat-toolbar color="primary" class="mat-elevation-z5">    <!-- Título -->
        <b>Interferometría</b>: Aplicación De Control Y Medición
    </mat-toolbar>

    <mat-tab-group>        <!-- Pestañas -->

        <mat-tab label="Configuración">        <!-- CONFIGURACIÓN -->
            <mat-card style="margin:10px">
                <mat-icon>settings</mat-icon>
                <h4>CONFIGURACIÓN</h4>
                Sigue los pasos indicados para configurar el sistema.
                <mat-horizontal-stepper [linear]="true" #stepper="matHorizontalStepper">

                    <mat-step>        <!-- Conexión -->
                        <ng-template matStepLabel>Conexión</ng-template>
                        <mat-card style="margin:10px">
                            <h4>Conexión del microcontrolador</h4>
                            Establece la conexión con el microcontrolador mediante un puerto serie
                            <br/><br/>
                            <span *ngIf="!puertoMicrocontrolador" style="color:crimson">No conectado...</span>
                            <span *ngIf="puertoMicrocontrolador" style="color:limegreen">¡Conectado!</span>
                            <br/><br/>
                            <mat-form-field>
                                <mat-select placeholder="Puerto" [(value)]= "puerto"
                                (selectionChange)="conectarMicrocontrolador()">
                                    <mat-option *ngFor="let puerto of puertosDisponibles"
                                    [value]="puerto.comName">
                                        {{puerto.comName}}
                                    </mat-option>
                                </mat-select>
                            </mat-form-field>
                        </mat-card>
                        <button mat-button matStepperNext>Siguiente</button>
                    </mat-step>

                    <mat-step>        <!-- Parámetros -->
                        <ng-template matStepLabel>Parámetros</ng-template>
                        <mat-card style="margin:10px">
                            <h4>Parámetros de muestreo</h4>
```

Selecciona la lambda del láser, la frecuencia de modulación y el número de muestras de adquisición

```

<br/><br/>
<span *ngIf="!puertoMicrocontrolador" style="color:crimson">No conectado al
microcontrolador. Haz clic en <em/>Conexión</em></span>
<div *ngIf="puertoMicrocontrolador">
  <b>Lambda/n = </b>
  <mat-form-field>
    <input #message matInput type="number" maxlength="4" [(ngModel)]="lambda"
(change)="cambiarLambda()"/>
  </mat-form-field>
  nm
  <br/>
  <b>Fmodulación = </b>
  {{frecuencia}} Hz
  <mat-slider [disabled]="false" [invert]="false" [max]="frecuenciaMaxima"
[min]="0" [step]="1000" [thumbLabel]="true" [tickInterval]="0" [(ngModel)]="frecuencia"
[vertical]="false" (change)="cambiarFrecuencia()"/></mat-slider>
  <br/>
  <b>Nmuestras = </b>
  <mat-form-field>
    <mat-select [(value)]="muestras" (selectionChange)="cambiarMuestras()">
      <mat-option *ngFor="let muestras of [64,128,256,512,1024,2048,4096,8192]"
[value]="muestras">
        {{muestras}}
      </mat-option>
    </mat-select>
  </mat-form-field>
</div>
</mat-card>
<button mat-button matStepperPrevious>Anterior</button>
<button mat-button matStepperNext>Siguiete</button>
</mat-step>

<mat-step>    <!--Adquisición-->
<ng-template matStepLabel>Adquisición</ng-template>
<mat-card style="margin:10px">
  <h4>Ajuste de ganancia en adquisición</h4>
  Haz clic en <em>Rango</em> y sigue las instrucciones hasta conseguir un rango
óptimo

  <br/><br/>
  <span *ngIf="!puertoMicrocontrolador" style="color:crimson">No conectado al
microcontrolador. Haz clic en <em/>Conexión</em></span>
  <div *ngIf="puertoMicrocontrolador">
    <span *ngIf="optimo==0" style="color:crimson">Rango indefinido...</span>
    <span *ngIf="optimo==1" style="color:goldenrod">Ajusta el rango girando el
potenciómetro 1</span>
    <span *ngIf="optimo==2" style="color:limegreen">¡Rango óptimo!</span>
  <br/><br/>

```



```

        <button mat-fab color="warn" (click)="enviarUSART('r')">Rango</button>
        <br/><br/>
        <div *ngIf="rango">
            <mat-form-field><input matInput placeholder="Mínimo (%)" disabled
[value]="rango[0]" style="color:black"></mat-form-field>
            <mat-form-field><input matInput placeholder="Máximo (%)" disabled
[value]="rango[1]" style="color:black"></mat-form-field>
            <mat-form-field><input matInput placeholder="Diferencia (%)" disabled
[value]="rango[1]-rango[0]" style="color:black"></mat-form-field>
        </div>
    </div>
</mat-card>
<button mat-button matStepperPrevious>Anterior</button>
<button mat-button matStepperNext>Siguiete</button>
</mat-step>

<mat-step>    <!--Calibración-->
    <ng-template matStepLabel>Calibración</ng-template>
    <mat-card style="margin:10px">
        <h4>Ajuste de amplitud de modulación</h4>
        Haz clic en <em>Auto</em> para calibrar el sistema automáticamente o utiliza los
        controles manuales <mat-icon>arrow_downward</mat-icon> y <mat-icon>arrow_upward</mat-icon>
        <br/><br/>
        <span *ngIf="!puertoMicrocontrolador" style="color:crimson">No conectado al
        microcontrolador. Haz clic en <em/>Conexión</em></span>
        <div *ngIf="puertoMicrocontrolador">
            <span *ngIf="calibrado==0" style="color:crimson">No calibrado...</span>
            <span *ngIf="calibrado==1" style="color:dodgerblue">Espera... ({{cuenta}}
segundos)</span>
            <span *ngIf="calibrado==2" style="color:goldenrod">Aumenta la ganancia girando
el potenciómetro 2 (Error = {{error}} %)</span>
            <span *ngIf="calibrado==3" style="color:limegreen">¡Calibrado! (Error =
{{error}} %)</span>
            <br/><br/>
            <button mat-fab color="warn" (click)="autocalibrar()">Auto</button>
            <br/>
            <mat-icon>warning</mat-icon> <em>Puede tardar hasta
{{{(30*tEstabilizacion)/1000}} segundos</em>
            <br/><br/>
            <b>Amplitud</b>:
            <button mat-mini-fab color="primary" (click)="enviarUSART('vd')"><mat-
icon>arrow_downward</mat-icon></button>
            {{amplitud}} / 30
            <button mat-mini-fab color="primary" (click)="enviarUSART('vi')"><mat-
icon>arrow_upward</mat-icon></button>
        </div>
    </mat-card>
    <button mat-button matStepperPrevious>Anterior</button>

```

```

        <button mat-button matStepperNext (click)="stepper.reset()">Terminar /
Volver</button>
    </mat-step>

</mat-horizontal-stepper>
</mat-card>
</mat-tab>

<mat-tab label="Medición">    <!--MEDICIÓN-->
    <mat-card style="margin:10px">
        <mat-icon>filter_center_focus</mat-icon>
        <h4>MEDICIÓN</h4>
        Haz clic en <em>Adquirir</em> para realizar una nueva medición y después haz clic en
        <em>Medida</em> o en <em>Detalles</em> para visualizar los resultados. Puedes configurar la
        adquisición cada cierto intervalo de tiempo activando <em>Adquisición programada</em>.
        <br/><br/>
        <span *ngIf="!puertoMicrocontrolador" style="color:crimson">No conectado al
        microcontrolador. Haz clic en <em>Configuración</em></span>
        <div *ngIf="puertoMicrocontrolador">
            <em>Adquirir automáticamente cada:</em>
            <br/>
            <div *ngIf="!contadorAdquisicion">
                <mat-select [(value)]="tiempoProgramado">
                    <mat-option *ngFor="let tiempo of [1,10,15,30,60]" [value]="tiempo">
                        {{tiempo}} minutos
                    </mat-option>
                </mat-select>
            </div>
            <span *ngIf="contadorAdquisicion"><b>{{tiempoProgramado}} minutos</b></span>
            <br/>
            <mat-slide-toggle (change)="toggleAdquisicionProgramada($event)">Adquisición
            programada</mat-slide-toggle>
            <mat-grid-list cols="3">
                <mat-grid-tile>
                    <button mat-raised-button (click)="adquirir()"><mat-icon>autorenew</mat-
                    icon><br/>Adquirir</button>
                </mat-grid-tile>
                <mat-grid-tile>
                    <button mat-raised-button (click)="mostrarGrafico('f')"><mat-icon>equalizer</mat-
                    icon><br/>Medida</button>
                </mat-grid-tile>
                <mat-grid-tile>
                    <button mat-raised-button [matMenuTriggerFor]="menu"><mat-icon>attach_file</mat-
                    icon><br/>Detalles</button>
                </mat-grid-tile>
                <mat-menu #menu>
                    <button mat-menu-item (click)="mostrarGrafico('m')">Muestras</button>
                    <button mat-menu-item (click)="mostrarGrafico('s')">Seno</button>
                    <button mat-menu-item (click)="mostrarGrafico('c')">Coseno</button>

```

```

        <button mat-menu-item (click)="mostrarGrafico('p')">Fase</button>
        <button mat-menu-item (click)="mostrarGrafico('d')">Desplazamiento</button>
        <button mat-menu-item (click)="mostrarGrafico('f')">FFT</button>
    </mat-menu>
</mat-grid-list>
<em>Última adquisición: </em>
<br/>
<span style="color:darkslateblue">{{fecha}}</span>
</div>
</mat-card>
</mat-tab>

<mat-tab label="Historial">      <!--HISTORIAL-->
    <mat-card style="margin:10px">
        <mat-icon>history</mat-icon>
        <h4>HISTORIAL</h4>
        Haz clic en <em>Importar</em> / <em>Exportar</em> para cargar/crear un archivo de
registros (<b>REGISTROS.json</b>), o en <mat-icon>equalizer</mat-icon> / <mat-icon>delete</mat-
icon> para ver/eliminar los resultados de un registro guardado y cargado.
        <br/><br/>
        <button mat-raised-button (click)="leerRegistros()">Importar</button>
        <button mat-raised-button color="primary"
(click)="escribirRegistros()">Exportar</button>
        <br/><br/>
        <mat-table #table [dataSource]="fuenteTabla" matSort class="mat-elevation-z5"
style="height:300px;overflow:auto">
            <ng-container matColumnDef="fecha">
                <mat-header-cell *matHeaderCellDef mat-sort-header>Fecha</mat-header-cell>
                <mat-cell *matCellDef="let registro">{{registro.fecha}}</mat-cell>
            </ng-container>
            <ng-container matColumnDef="ver">
                <mat-header-cell *matHeaderCellDef mat-sort-header>Ver</mat-header-cell>
                <mat-cell *matCellDef="let registro"><button mat-button color="primary"
(click)="mostrarRegistro(registro)"><mat-icon>equalizer</mat-icon></button></mat-cell>
            </ng-container>
            <ng-container matColumnDef="eliminar">
                <mat-header-cell *matHeaderCellDef mat-sort-header>Eliminar</mat-header-cell>
                <mat-cell *matCellDef="let registro"><button mat-button color="warn"
(click)="eliminarRegistro(registro)"><mat-icon>delete</mat-icon></button></mat-cell>
            </ng-container>
            <mat-header-row *matHeaderRowDef=["fecha","ver","eliminar"]></mat-header-row>
            <mat-row *matRowDef="let row; columns: ['fecha','ver','eliminar'];"></mat-row>
        </mat-table>
    </mat-card>
</mat-tab>

</mat-tab-group>
</div>

```

APP.COMPONENT.TS

```
/*CONSTANTES*/

const LAMBDA: number = 633;
const FRECUENCIA: number = 20000;
const MUESTRAS: number = 2048;
const T_ESTABILIZACION: number = 1000;

/*CLASE DE UN REGISTRO*/

export class registro {
    fecha: string;
    muestras: number;
    frecuencia: number;
    fft: number[];
}

/*MÓDULOS EXTERNOS*/

/*Node.js*/

var serialPort = require('serialport');    //Módulo de comunicación por puertos serie
var fs = require('fs');    //Módulo para la lectura y escritura de archivos

/*Angular*/

import { NgModule, Component, AfterViewInit, ViewChild, Inject, ViewEncapsulation } from
'@angular/core';    //Módulos de Angular
import { BrowserModule } from '@angular/platform-browser';

import { MatSliderModule } from '@angular/material/slider';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';    //Módulos de Material
import { MatInputModule } from '@angular/material/input';
import { MatToolbarModule } from '@angular/material/toolbar';
import { MatCardModule } from '@angular/material/card';
import { MatTabsModule } from '@angular/material/tabs';
import { MatSelectModule } from '@angular/material/select';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { MatButtonModule } from '@angular/material/button';
import { MatIconModule } from '@angular/material/icon';
import { MatSnackBar, MatSnackBarModule, MatTableDataSource } from '@angular/material';
import { MatStepperModule } from '@angular/material/stepper';
import { MatDialogModule, MatDialog, MatDialogRef, MAT_DIALOG_DATA } from
'@angular/material/dialog';
import { MatMenuModule } from '@angular/material/menu';
import { MatGridListModule } from '@angular/material/grid-list';
import { MatTableModule } from '@angular/material/table';
import { MatSlideToggleModule, MatSlideToggleChange } from '@angular/material/slide-toggle';
```

```

import 'hammerjs';          //Módulos de complementos
import {Chart} from 'chart.js';

/*COMPONENTES Y MÓDULO DE LA APLICACIÓN*/

/*Componente raíz de la aplicación*/

@Component({
  selector: 'App',
  templateUrl: 'app.component.html',
  encapsulation: ViewEncapsulation.None
})
export class AppComponent {
  @ViewChild('grafico') contexto: any;
  constructor(private estado: MatSnackBar, private dialogo: MatDialog) {
    this.leerRegistros();
  }

  historial: registro[] = [];          //Variable que contiene el historial de las mediciones
  fuenteTabla = new MatTableDataSource<registro>(this.historial);

  buscarPuertos: any = window.setInterval(() => {          //Variables para la comunicación con el
microcontrolador
    serialPort.list((err: any, res: any) => {
      if(!err) {
        this.puertosDisponibles = res;
      }
    })
  }, 1000);
  puertosDisponibles: any;
  puerto: string;
  puertoMicrocontrolador: any;
  parser=new serialPort.parsers.Readline({ delimiter: '\r\n' });

  readonly frecuenciaMaxima: number = FRECUENCIA;          //Variables de control del sistema
  readonly tEstabilizacion: number = T_ESTABILIZACION;
  rango: number[];
  optimo: number = 0;
  cuenta: number;
  contadorAutocalibracion: any;
  calibrado: number = 0;
  fecha: string = "Nunca";
  lambda: number = LAMBDA;
  frecuencia: number = FRECUENCIA;
  muestras: number = MUESTRAS;
  amplitud: number = 15;
  error: number;
  vout: number[];

```

```

seno: number[];
coseno: number[];
fase: number[];
desplazamiento: number[];
fft: number[];
tiempoProgramado: number = 15;
contadorAdquisicion: any;

conectarMicrocontrolador() { //Función para la conexión y la configuración de la
comunicación con el microcontrolador
    this.puertoMicrocontrolador = new serialPort(this.puerto, {
        baudRate: 115200,
    });
    this.puertoMicrocontrolador.pipe(this.parser);
    this.puertoMicrocontrolador.on('open', () => {
        window.clearInterval(this.buscarPuertos);
        this.estado.open("Conectado al microcontrolador: " + this.puerto, "Cerrar", { duration:
2000 }) ;
        this.enviarUSART('i')
    })
    this.parser.on('data', (dato: string) => {
        //console.log("Dato recibido del microcontrolador: " + dato);
        switch(dato.substring(0,1)) {
            case 'm': //Muestras
                this.vout=this.recibirArrayUSART(dato.substring(1));
                this.enviarUSART('gs');
                break;
            case 's': //Seno
                this.seno=this.recibirArrayUSART(dato.substring(1));
                this.enviarUSART('gc');
                break;
            case 'c': //Coseno
                this.coseno=this.recibirArrayUSART(dato.substring(1));
                this.enviarUSART('gp');
                break;
            case 'p': //Fase
                this.fase=this.recibirArrayUSART(dato.substring(1));
                this.enviarUSART('gd');
                break;
            case 'd': //Desplazamiento
                this.desplazamiento=this.recibirArrayUSART(dato.substring(1));
                this.enviarUSART('gf');
                break;
            case 'f': //FFT
                this.fft=this.recibirArrayUSART(dato.substring(1));
                this.fecha=this.registrarFecha();
                this.historial.push({"fecha": this.fecha, "muestras": this.muestras, "frecuencia":
this.frecuencia, "fft": this.fft});
                this.fuenteTabla = new MatTableDataSource<registro>(this.historial);

```

```

        this.estado.open("¡Adquisición completada!", "Cerrar", { duration: 2000 });
        break;
    case 'r': //Rango
        this.rango=this.recibirArrayUSART(dato.substring(1));
        if(this.rango[0] < 15 && this.rango[0] > 3 && this.rango[1] > 85 && this.rango[1] <
97) {
            this.optimo=2;
        } else {
            this.optimo=1;
        }
        break;
    case 'a': //Autocalibración
        var resultado = this.recibirArrayUSART(dato.substring(1));
        if(resultado[0] != -1) {
            this.calibrado=3;
            this.amplitud=resultado[0];
            this.error=resultado[1];
        } else {
            this.calibrado=2;
            this.amplitud=30;
            this.error=resultado[1];
        }
        window.clearInterval(this.contadorAutocalibracion);
        break;
    case 'v': //Voltaje de amplitud
        this.amplitud=this.recibirArrayUSART(dato.substring(1))[0];
        this.calibrado=3;
        break;
    case 'i': //Información
        this.estado.open('INFORMACIÓN: ' + dato.substring(1), "Cerrar", { duration: 2000 });
        break;
    case 'e': //Error
        this.estado.open('ERROR: ' + dato.substring(1), "Cerrar", { duration: 2000 });
        break;
    default:
    }
    })
    this.puertoMicrocontrolador.on('error', (err: any) => {
        this.estado.open("ERROR: No conectado al microcontrolador", "Cerrar", { duration: 2000 });
        this.puertoMicrocontrolador = null;
        console.log(err);
    })
    }
    enviarUSART(dato: any) { //Función para el envío de datos mediante USART
        this.puertoMicrocontrolador.write(dato, 'ascii');
        //console.log("Dato enviado al microcontrolador: " + dato);
    }
    recibirArrayUSART(cadena: string) { //Función para la recepción de arrays de números a
través de USART

```

```

var array: number[] = [];
for(var i = 0; i < cadena.length; i++) {
    if(cadena[i] != '{' && cadena[i] != ',') {
        for(var j = i+1; j < cadena.length; j++) {
            if(cadena[j] == ',' || cadena[j] == '}')
            {
                array.push(Number(cadena.substring(i,j)))
                i=j+1;
            }
        }
    }
}
return array;
}

cambiarLambda() {    //Función para modificar la lambda del láser
    var buffer = Buffer.alloc(2);
    buffer.writeUInt16LE(this.lambda, 0);
    this.enviarUSART('l');
    this.enviarUSART(buffer);
}

cambiarFrecuencia() {    //Función para modificar la frecuencia de modulación
    var buffer = Buffer.alloc(2);
    buffer.writeUInt16LE(this.frecuencia, 0);
    this.enviarUSART('f');
    this.enviarUSART(buffer);
}

cambiarMuestras() {    //Función para modificar el número de muestras
    var buffer = Buffer.alloc(2);
    buffer.writeUInt16LE(this.muestras, 0);
    this.enviarUSART('n');
    this.enviarUSART(buffer);
}

autocalibrar() {    //Función para solicitar una rutina de autocalibración
    this.contadorAutocalibracion = window.clearInterval(this.contadorAutocalibracion);
    this.contadorAutocalibracion = null;
    this.enviarUSART('a');
    this.calibrado = 1;
    this.cuenta = (30 * T_ESTABILIZACION) / 1000;
    this.contadorAutocalibracion = window.setInterval(() => {
        if(this.cuenta > 0) {
            this.cuenta--;
        }
    }, 1000);
}

adquirir() {    //Función para realizar una nueva medición
    this.enviarUSART('m');
    this.estado.open("Adquiriendo...", "Cerrar", { duration: 2000 });
}

```



```

toggleAdquisicionProgramada(event: MatSlideToggleChange) { //Función para activar o
desactivar la adquisición continua programada
    if (event.checked == true) {
        this.contadorAdquisicion = window.clearInterval(this.contadorAdquisicion);
        this.contadorAdquisicion = window.setInterval(() => {
            this.adquirir()
        }, this.tiempoProgramado * 60 * 1000);
        this.estado.open("Adquisición programada activada (cada " + this.tiempoProgramado + "
minutos)", "Cerrar", { duration: 2000 });
    } else {
        this.contadorAdquisicion = window.clearInterval(this.contadorAdquisicion);
        this.contadorAdquisicion = null;
        this.estado.open("Adquisición programada desactivada", "Cerrar", { duration: 2000 });
    }
}

mostrarGrafico(opcion: string) { //Función para mostrar los gráficos actuales
    var etiquetas: string[] = [];
    switch(opcion) {
        case 'm':
            for(var i = 0; i < this.vout.length; i++) {
                etiquetas.push(((i*1/(this.frecuencia*2))*1000).toFixed(1));
            }
            this.dialogo.open(Grafico, { data: { ejex: 't [ms]', ejey: 'Muestras [ADC Sin Signo]',
x: etiquetas, y: this.vout, color: 'rgb(63, 81, 181)', titulo: 'Muestras' }}});
            break;
        case 's':
            for(var i = 0; i < this.seno.length; i++) {
                etiquetas.push(((i*1/this.frecuencia+1/(this.frecuencia*2))*1000).toFixed(1));
            }
            this.dialogo.open(Grafico, { data: { ejex: 't [ms]', ejey: 'Seno [ADC Con Signo]', x:
etiquetas, y: this.seno, color: 'rgb(33, 150, 243)', titulo: 'Seno' }}});
            break;
        case 'c':
            for(var i = 0; i < this.coseno.length; i++) {
                etiquetas.push(((i*1/this.frecuencia+1/(this.frecuencia*2))*1000).toFixed(1));
            }
            this.dialogo.open(Grafico, { data: { ejex: 't [ms]', ejey: 'Coseno [ADC Con Signo]', x:
etiquetas, y: this.coseno, color: 'rgb(244, 67, 54)', titulo: 'Coseno' }}});
            break;
        case 'p':
            for(var i = 0; i < this.fase.length; i++) {
                etiquetas.push(((i*1/this.frecuencia+1/(this.frecuencia*2))*1000).toFixed(1));
            }
            this.dialogo.open(Grafico, { data: { ejex: 't [ms]', ejey: 'Fase [rad]', x: etiquetas,
y: this.fase, color: 'rgb(255, 193, 7)', titulo: 'Fase' }}});
            break;
        case 'd':
            for(var i = 0; i < this.desplazamiento.length; i++) {
                etiquetas.push(((i*1/this.frecuencia+1/(this.frecuencia*2))*1000).toFixed(1));
            }
    }
}

```

```

    }
    this.dialogo.open(Grafico, { data: { ejex: 't [ms]', ejey: 'Desplazamiento [nm]', x:
etiquetas, y: this.desplazamiento, color: 'rgb(76, 175, 80)', titulo: 'Desplazamiento' }});
    break;
    case 'f':
        for(var i = 0; i < this.fft.length; i++) {
            etiquetas.push((i*this.frecuencia/(this.muestras/2)).toFixed(0));
        }
        this.dialogo.open(Grafico, { data: { ejex: 'f [Hz]', ejey: 'Desplazamiento [nm]', x:
etiquetas, y: this.fft, color: 'rgb(96, 125, 139)', titulo: 'FFT' }});
        break;
    default::;
}
}
}
leerRegistros() { //Función para leer los registros de un archivo
    this.estado.open("Leyendo registros...", "Cerrar", { duration: 2000 });
    fs.readFile('REGISTROS.json', (err: any, dat: any) => {
        if(err) {
            this.estado.open("ERROR: No se pudo leer el archivo", "Cerrar", { duration: 2000 });
        } else {
            try {
                this.historial = JSON.parse(dat);
                this.fuenteTabla = new MatTableDataSource<registro>(this.historial);
                this.estado.open("¡Registros importados desde 'REGISTROS.json'!", "Cerrar", {
duration: 2000 });
            } catch(e) {
                this.estado.open("ERROR: No se reconoció el formato JSON en el contenido del archivo",
"Cerrar", { duration: 2000 });
            }
        }
    });
}
}
escribirRegistros() { //Función para escribir los registros actuales en un archivo
    this.estado.open("Escribiendo registros...", "Cerrar", { duration: 2000 });
    fs.writeFile('REGISTROS.json', JSON.stringify(this.historial, null, 2), (err: any) => {
        if(err) {
            this.estado.open("ERROR: No se pudo escribir el archivo", "Cerrar", { duration: 2000 });
        } else {
            this.estado.open("¡Registros exportados a 'REGISTROS.json'!", "Cerrar", { duration: 2000
});
        }
    });
}
}
eliminarRegistro(datos: registro) { //Función para eliminar un registro
    for(var i = 0; i < this.historial.length; i++) {
        if(this.historial[i].fecha == datos.fecha) {
            this.historial.splice(i, 1);
        }
    }
}
}

```

```

        this.fuenteTabla = new MatTableDataSource<registro>(this.historial);
        this.estado.open("Registro eliminado: " + datos.fecha, "Cerrar", { duration: 2000 });
    }

    mostrarRegistro(datos: registro) {        //Función para mostrar el gráfico de los datos de un
registro
        var etiquetas: string[] = [];
        for(var i = 0; i < datos.fft.length; i++) {
            etiquetas.push((i*datos.frecuencia/(datos.muestras/2)).toFixed(0));
        }
        this.dialogo.open(Grafico, { data: { ejex: 'f [Hz]', ejey: 'Desplazamiento [nm]', x:
etiquetas, y: datos.fft, color: 'rgb(96, 125, 139)', titulo: 'FFT [' + datos.fecha + ']' });
    }

    registrarFecha() {        //Función para obtener la fecha y hora en formato de cadena de
caracteres
        var cadena: string = "";
        var fecha: Date = new Date();
        cadena = fecha.getHours().toString();
        cadena = cadena + ':';
        if(fecha.getMinutes() < 10) {
            cadena = cadena + '0' + fecha.getMinutes().toString()
        } else {
            cadena = cadena + fecha.getMinutes().toString()
        }
        cadena = cadena + ' ('
        cadena = cadena + fecha.getDate().toString()
        cadena = cadena + '/'
        cadena = cadena + (fecha.getMonth()+1).toString()
        cadena = cadena + '/'
        cadena = cadena + fecha.getFullYear().toString()
        cadena = cadena + ')'
        return cadena;
    }
}

/*Componente general para generar cada gráfico*/

@Component({
    selector: 'grafico',
    template: `
        <h1 mat-dialog-title>Gráfico De <em>{{titulo}}</em></h1>
        <div mat-dialog-content>
            <canvas #grafico height="400px" width="600px" style="z-index: 1"></canvas>
        </div>
        <div mat-dialog-actions>
            <button mat-raised-button (click)="cerrar()">Cerrar</button>
        </div>
    `,
    encapsulation: ViewEncapsulation.None
})

```

```

export class Grafico implements AfterViewInit {
  @ViewChild('grafico') contexto: any;
  constructor(private dialogo: MatDialogRef<Grafico>, @Inject(MAT_DIALOG_DATA) private data:
any) {}
  titulo: string = this.data.titulo;
  grafico: any;
  ngAfterViewInit() {
    this.grafico = new Chart(this.contexto.nativeElement.getContext('2d'), {
      type: 'line',
      data: {
        labels: this.data.x,
        datasets: [{
          data: this.data.y,
          backgroundColor: this.data.color,
          borderColor: this.data.color,
          fill: false,
        }]
      },
      options: {
        responsive: true,
        maintainAspectRatio: true,
        legend: {
          display: false
        },
        scales: {
          xAxes: [{
            display: true,
            scaleLabel: {
              display: true,
              labelString: this.data.ejex
            },
            ticks: {
              autoSkip: true,
              maxTicksLimit: 10,
              maxRotation: 0,
            }
          }],
          yAxes: [{
            display: true,
            scaleLabel: {
              display: true,
              labelString: this.data.ejey
            }
          }]
        }
      }
    });
  }
  cerrar(): void {

```

```

        this.dialogo.close();
    }
}

/*Módulo raíz de la aplicación*/

@NgModule({
  imports: [
    BrowserModule,
    BrowserAnimationsModule,
    FormsModule,
    ReactiveFormsModule,
    MatButtonModule,
    MatInputModule,
    MatToolbarModule,
    MatTabsModule,
    MatCardModule,
    MatSnackBarModule,
    MatDialogModule,
    MatIconModule,
    MatSliderModule,
    MatStepperModule,
    MatMenuModule,
    MatGridListModule,
    MatSelectModule,
    MatTableModule,
    MatSlideToggleModule
  ],
  declarations: [
    AppComponent,
    Grafico
  ],
  bootstrap: [AppComponent],
  entryComponents: [
    Grafico
  ]
})
export class AppModule { }

```

ANEXO V: RESUMEN PRESENTADO EN EL SAAEI 2018

Sistema electrónico para interferometría con detección síncrona

Alejandro J. Ramos San Juan, Dragos A. Poiana y José A. García Souto

Departamento de Tecnología Electrónica, GOTL, Univ. Carlos III de Madrid, Butarque 15, 28911 Leganés

jsouto@ing.uc3m.es

Resumen— Los sensores interferométricos de fibra óptica tienen muy alta sensibilidad para la detección de vibraciones, emisiones acústicas y ultrasonidos. Sin embargo, el sistema de detección de la fase óptica puede ser complejo o difícil de implementar fuera de un laboratorio. En este trabajo se presenta una solución mediante detección síncrona. El sistema electrónico desarrollado modula la fase óptica a frecuencia de kHz y proporciona una detección equivalente I-Q que permite detectar sin ambigüedad la medida realizada por el sensor interferométrico a pesar de su función de transferencia no lineal y periódica.

La implementación realizada se ha aplicado a la medida de vibraciones de transformadores, pero puede utilizarse a frecuencias de audio y sónica. La misma técnica síncrona puede aplicarse a esquemas de interferencia para medida sin contacto y la modulación puede realizarse sobre el láser a frecuencias de MHz para detectar ultrasonidos.

Palabras clave—sensor de fibra óptica, interferómetro láser, detección síncrona, muestreo síncrono, vibraciones

1. INTRODUCCIÓN

EN la actualidad existe la necesidad de monitorizar el comportamiento de máquinas y equipamientos con el objetivo de conocer su estado y su correcto funcionamiento. Las vibraciones y su evolución se utilizan a menudo como fuente de información para el diagnóstico. Sin embargo, muchos sensores tienen limitaciones en los ambientes hostiles (temperatura, radiación electromagnética, etc.), a diferencia de lo que ocurre con los sensores ópticos y de fibra óptica.

Por otra parte, mediante el uso de la interferometría óptica se puede lograr un gran ancho de banda de detección de frecuencias de vibración y una alta sensibilidad en comparación con otras técnicas utilizadas [1].

Se han utilizado vibrómetros sin contacto con luz hacia una superficie [2]. También se pueden medir vibraciones con sensores de fibra óptica basados en acelerómetros de resorte, [3-5] o voladizos [6,7], pero tienen limitaciones en la sensibilidad y en la respuesta dinámica.

El sistema de medida propuesto se basa en un esquema de interferometría láser que detecta la deformación o el desplazamiento inducidos por las vibraciones mecánicas en un cabezal sensor de fibra óptica. Aunque existen otros

transductores en fibra (redes de Bragg y cavidades de Fabry-Perot), la fibra óptica es intrínsecamente más sensible, si la longitud es suficiente y se usa un interferómetro láser [8]. Por ejemplo, la lectura de fase se relaciona con la longitud de onda óptica, por lo tanto 2π rad corresponde a $1,5 \mu\text{m}$ típicamente.

En cuanto a la implementación, se busca robustez, bajo coste, simplicidad y escalabilidad para el cabezal sensor, el interferómetro óptico y el procesamiento de fase óptica. En este sentido, como el sistema de interferencia tiene una función de transferencia no lineal en coseno alzado, los mayores esfuerzos se centran en la demodulación. Se utilizan esquemas de compensación activa de la deriva de fase, incluso con la frecuencia del láser [9], demodulación con dos señales en cuadratura [10], esquemas heterodinos y pseudo-heterodinos [11,12], actualmente con implementación digital [13]. Todos estos casos tienen gran complejidad en la electrónica de acondicionamiento, o bien dependen de ajustes de calibración en cada tanda de medidas.

Los esquemas con mayor potencial en la detección de fase óptica son los que proporcionan dos salidas en cuadratura (seno/coseno); detección I-Q, por analogía en comunicaciones. Los más usados están basados en PGC (*Phase Generated Carrier*) [14], los cuales generan dos armónicos de frecuencia que proporcionan ambas señales en cuadratura. Sin embargo, presentan distorsión armónica (no linealidad en la medida) y alta dependencia de la relación de amplitudes, particularmente en implementaciones digitales y micro-programadas.

En este trabajo proponemos un esquema alternativo con dos salidas en cuadratura obtenidas a partir de detección síncrona. Anteriormente hemos utilizado esquemas análogos para separar dos magnitudes de distinta frecuencia [15]. En este caso, una de las magnitudes es una portadora forzada y en sincronismo, que además facilita la normalización de amplitudes y facilita una implementación sencilla con circuitos digitales o con sistemas de adquisición programables.

En segundo lugar, como ejemplo de aplicación se compara el sensor de fibra óptica y detección síncrona con un acelerómetro piezoeléctrico, midiendo las vibraciones del núcleo de un transformador de baja potencia. El sensor de fibra óptica no sólo supera al acelerómetro en inmunidad electromagnética y aislamiento, si no también en sensibilidad y precisión.

II. PRINCIPIO DE MEDIDA

A. Interferómetro con modulación de la fase óptica

El sistema óptico usado para realizar las medidas es un interferómetro de Mach-Zehnder mostrado en la Fig. 1. En esta configuración se utiliza un láser, dos acopladores de fibra óptica (uno como divisor óptico y otro como recombinación) y un piezoeléctrico (PZT), cuyo propósito se explica más adelante.



Fig. 1. Interferómetro de Mach-Zehnder. I_0 es la intensidad óptica del láser, V es la visibilidad y ϕ es la fase óptica.

La medida de las vibraciones mecánicas se basa fundamentalmente en detectar la elongación sobre los brazos de fibra óptica que se deriva de ellas. Esta pequeña elongación se obtiene a través de la fase óptica de la luz que atraviesa la fibra, cuyo valor varía linealmente con el desplazamiento (1):

$$\phi_m(t) - \phi_r = \frac{2\pi}{\lambda} |L_m(t) - L_r| \quad (1)$$

donde:

$\phi_m(t)$	fase óptica de medida;
ϕ_r	fase óptica de referencia;
λ	longitud de onda del láser;
$L_m(t)$	longitud del brazo de medida;
L_r	longitud del brazo de referencia.

En este caso se va a utilizar una técnica de detección sincrónica. Consiste en modular la fase óptica mediante una frecuencia alta (frecuencia de la señal portadora) que posteriormente permitirá extraer la información de frecuencias más bajas objeto de medida (frecuencias de vibración). La introducción de la portadora en el sistema se realiza excitando con cierta tensión el PZT del esquema de la Fig. 1.

A partir del muestreo sincrónico en los máximos y mínimos de esta señal portadora de modulación en fase óptica, se obtienen dos señales cuyos valores responden a la función de transferencia del interferómetro (2) representada en la Fig. 2.

$$I(t) = I_0(1 + V \cdot \cos \phi(t)) \quad (2)$$

donde:

$I(t)$	intensidad óptica de salida del interferómetro;
I_0	intensidad óptica del láser;
V	visibilidad;
$\phi(t)$	fase óptica.

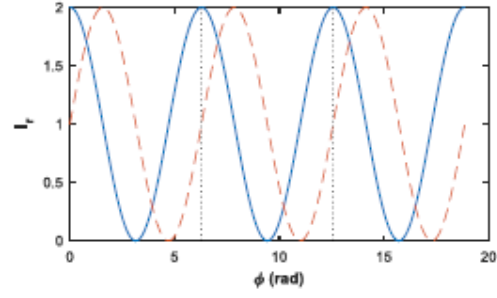


Fig. 2. Función de transferencia del interferómetro de Mach-Zehnder (azul) y su respuesta en cuadratura, con una diferencia de fase de $-\frac{\pi}{2}$ rad (naranja). I_r es la intensidad óptica normalizada ($1 + \cos \phi$ y $1 + \sin \phi$) y ϕ es la fase óptica.

Estas dos señales deben encontrarse en cuadratura, es decir, la diferencia de fase óptica entre los máximos y mínimos debe ser de $\frac{\pi}{2}$ rad (Fig. 2). De esta manera, por un lado, se obtiene la respuesta del interferómetro para una fase óptica determinada y por otro la respuesta para la misma fase menos $\frac{\pi}{2}$ rad, siendo equivalente a obtener la misma función de transferencia, pero cambiando el coseno por el seno.

Así se obtienen tres ventajas principales frente a otras técnicas. La primera ventaja es la mejora de la sensibilidad, ya que esta varía en función del punto de trabajo del interferómetro debido al carácter no lineal de su función de transferencia. Cuando una de las señales se encuentra en uno de los extremos donde las funciones seno y coseno pierden sensibilidad, la otra se encuentra en la zona de mayor sensibilidad. Por otro lado, se soluciona el problema de ambigüedad del signo, ya que a partir de las funciones seno y coseno obtenidas se puede determinar la fase de forma inequívoca y también el sentido de desplazamiento. Por último, mediante este método no es necesario mantener el interferómetro en un punto de trabajo fijo de forma activa [8].

B. Detección sincrónica de la fase óptica

La fase óptica total en el interferómetro corresponde a la suma de distintas contribuciones (3).

$$\phi = \phi_0 + \phi_m(t) + \phi_p(t) \quad (3)$$

donde:

ϕ_0	fase óptica inicial;
$\phi_m(t)$	fase óptica de medida;
$\phi_p(t)$	fase óptica portadora.

La fase óptica portadora viene dada por la frecuencia de la señal portadora con la que se excita el PZT (4).

$$\phi_p(t) = \phi_{p0} \cos(\omega_p t) = \phi_{p0} \cos(2\pi f_p t) \quad (4)$$

donde:

- $\phi_p(t)$ fase óptica portadora;
- ϕ_{p0} amplitud máxima de la fase óptica portadora;
- ω_p frecuencia angular de la señal portadora;
- f_p frecuencia de la señal portadora;

Muestrear la salida en los máximos y mínimos es equivalente a muestrear en los instantes de tiempo en los que $\omega_p t = \frac{\pi}{2} + 2\pi k$ (para los máximos) y $\omega_p t = \frac{3\pi}{2} + 2\pi k$ (para los mínimos) con $k = 0, 1, 2, \dots$, siendo por tanto necesario un valor de amplitud máxima $\phi_{p0} = \frac{\pi}{4}$ radianes. De esta manera, se obtienen las dos señales de salida en cuadratura correspondientes a los máximos (I_{Mk}) y a los mínimos (I_{mk}) en cada instante muestreado k (5).

$$\begin{aligned} I_{Mk} &= I_0(1 + V \cdot \cos(\phi_0 + \phi_m(k) + \phi_{p0})) \\ I_{mk} &= I_0(1 + V \cdot \cos(\phi_0 + \phi_m(k) - \phi_{p0})) \end{aligned} \quad (5)$$

Una vez obtenidas las medidas se procede a la demodulación de la fase óptica. Conociendo que una de las señales en cuadratura es el coseno y la otra es el seno, ambas con una componente continua y multiplicadas por la intensidad óptica del láser, se obtiene la tangente correspondiente a la fase en cada instante muestreado k (6).

$$\tan \phi(k) = \frac{I_0(1 + V \cdot \sin \phi(k)) - I_0}{I_0(1 + V \cdot \cos \phi(k)) - I_0} \quad (6)$$

La fase óptica $\phi(k)$ se obtiene por tanto aplicando la función arcotangente al resultado anterior y se compensa restando $\frac{\pi}{4}$ radianes debido a la modulación en fase óptica de $\pm \frac{\pi}{4}$ radianes. La fase óptica total acumulada (ϕ_a) se obtiene sumando las variaciones de fase obtenidas (7), siendo ϕ_{a0} la fase óptica acumulada inicial.

$$\begin{aligned} \phi(k) &= \arctan(\tan \phi(k)) - \frac{\pi}{4} \\ \phi_a &= \phi_{a0} + \sum_{k=0}^{k-1} \phi(k+1) - \phi(k) \end{aligned} \quad (7)$$

Aunque mediante este método se consigue la determinación exacta de la fase, existen puntos críticos a tener en cuenta. Estos puntos son aquellos en los que la tangente tiene un valor infinito, y por tanto la fase óptica vale $\pm \frac{\pi}{2}$ radianes. En estos puntos se produce una discontinuidad de $\frac{\pi}{2}$ a $-\frac{\pi}{2}$ o al revés, y por tanto se deben considerar para acumular el desplazamiento de fase óptica correctamente y obtener así una curva continua y determinada para cualquier instante de tiempo. Para ello se tiene en cuenta la variación máxima existente en valor absoluto entre los dos puntos de alrededor, es decir, π radianes, y se calcula la diferencia de fase de ajuste necesaria ($\Delta\phi_{AB(\frac{\pi}{2}+\frac{\pi}{2})}$ o $\Delta\phi_{AB(\frac{\pi}{2}-\frac{\pi}{2})}$) para que la suma de esta y la diferencia de fase entre estos dos puntos sea $\pm\pi$ radianes según corresponda (8).

$$\begin{aligned} \Delta\phi_{AB(\frac{\pi}{2}+\frac{\pi}{2})} &= \pi + (\phi_B - \phi_A) \\ \Delta\phi_{AB(\frac{\pi}{2}-\frac{\pi}{2})} &= -\pi + (\phi_B - \phi_A) \end{aligned} \quad (8)$$

III. SISTEMA DE INSTRUMENTACIÓN

El sistema de instrumentación diseñado para el fin propuesto se sintetiza en el diagrama de la Fig. 3.

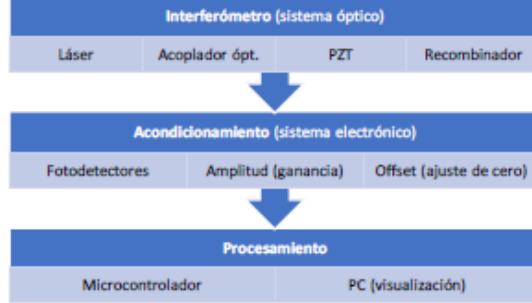


Fig. 3. Esquema de bloques del sistema de instrumentación completo

Consta de tres bloques principales: un interferómetro (sistema óptico), un acondicionamiento de las señales (sistema electrónico) y un procesamiento y visualización de resultados.

El primer bloque es un interferómetro de Mach-Zehnder en el que uno de los brazos se usa como referencia (Fig. 1) y está enrollado a un PZT permitiendo la modulación de la fase óptica al aplicarle tensión, mientras que el otro se usa para la medida de las vibraciones. El acoplador óptico de recombinación proporciona dos salidas de interferencia complementarias, las cuales se conectan al sistema electrónico de acondicionamiento.

En el sistema electrónico se detectan ambas señales de interferencia mediante dos foto-detectores conectados con fibra óptica. Mediante el uso de amplificadores de transimpedancia se obtiene a la salida una tensión proporcional a la diferencia de ambas señales, eliminado de esta forma la componente continua. La salida de tensión está acondicionada para ser adquirida y procesada por un microcontrolador que proporcionará los resultados de medida.

En el bloque de procesamiento se utiliza un microcontrolador en el que se adquiere la señal muestreando sincronamente y se procesan los datos adquiridos para posteriormente transferir los resultados obtenidos al ordenador y ser presentados.

IV. ANÁLISIS DEL SISTEMA DE DETECCIÓN SÍNCRONA

A continuación, se muestran los resultados obtenidos a partir de una excitación armónica de deformación sobre la fibra óptica, utilizada para comprobar el funcionamiento del muestreo sincrónico e ilustrar el procedimiento.

La señal de modulación varía entre $\pm \frac{\pi}{4}$ radianes, y su amplitud depende de la respuesta del PZT frente a la tensión aplicada. La frecuencia de la onda sinusoidal para excitar el PZT es de 20 kHz, evitando el reducido rango de frecuencias en las que se pueda manifestar el efecto de resonancia del PZT. Esta fase óptica portadora ϕ_p se representa en la Fig. 4a, con

los máximos y mínimos marcados, instantes en los que se muestrea la señal de salida del interferómetro.

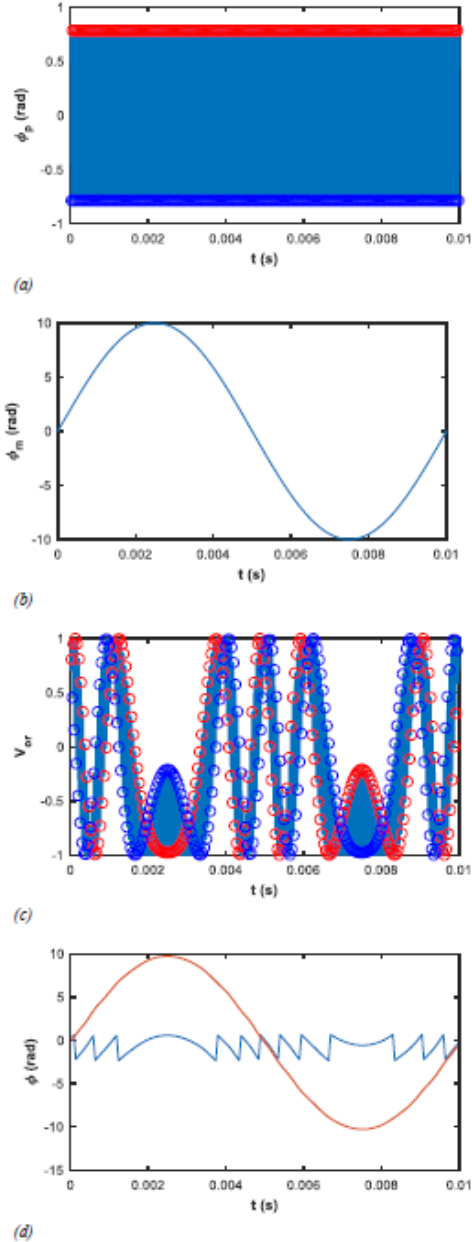


Fig. 4. Resultados de análisis del sistema, (a) portadora generada y marcas en los instantes de muestreo en máximos y mínimos, (b) señal de fase óptica simulada, (c) señales de interferencia en cuadratura como resultado del muestreo sincrónico, (d) fase instantánea y fase acumulada: reconstrucción de la fase óptica idéntica a la simulada en (b).

En la Fig. 4b se muestra como ejemplo una vibración en fase ϕ_m de 100 Hz objeto de medida, siendo la única perturbación detectada en el interferómetro (despreciando la deriva de fase debido a la temperatura, por ejemplo, que varía de forma mucho más lenta).

En la Fig. 4c se presenta el voltaje de salida relativo (V_{or}) muestreado tanto en los máximos como en los mínimos que conforman las dos señales de salida en cuadratura necesarias para demodular la fase óptica.

Aplicando los cálculos desarrollados anteriormente, se obtiene la fase óptica ϕ en la Fig. 4d (azul). Teniendo en cuenta la periodicidad de la función arcotangente y que la variación de fase absoluta máxima es mayor que 2π radianes, se acumula la fase comenzando desde cero (naranja).

En la Fig. 5 se muestra un detalle de la fase óptica portadora con los instantes de muestreo y de las señales de salida obtenidas en sincronismo, cuyo desfase relativo es $\pi/2$ rad.

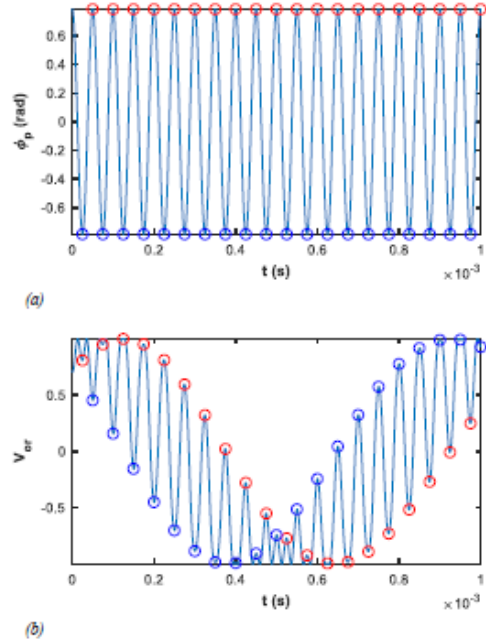


Fig. 5. Detalles de (a) la modulación en fase óptica y (b) la salida del interferómetro muestreada sincronamente.

V. RESULTADOS DE VIBRACIONES DE UN TRANSFORMADOR

Como prueba de concepto, se han realizado medidas con el sistema en un transformador cuyas vibraciones mecánicas se pretenden medir. En la Fig. 6 se muestra una fotografía de un transformador de laboratorio. Puede observarse un arrollamiento de fibra óptica como sensor fijado a la superficie del núcleo del transformador y un acelerómetro piezoeléctrico (Brüel&Kjaer) utilizado como referencia de comparación. La excitación del transformador se ha ajustado con un transformador variable para aplicar tensión nominal (220 V) y sobretensión. La frecuencia de vibraciones característica es 100

Hz (doble de la frecuencia de excitación), típicamente por efecto de la magnetostricción, y hasta la décima componente armónica (1 kHz) [1].

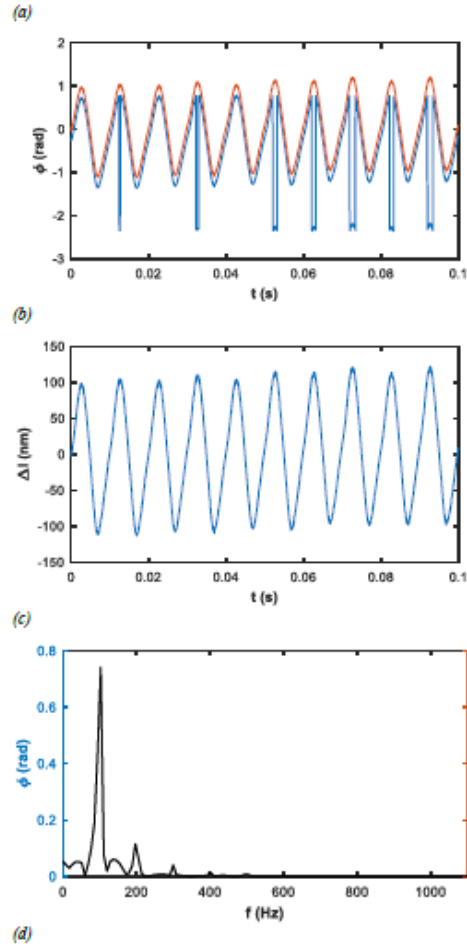
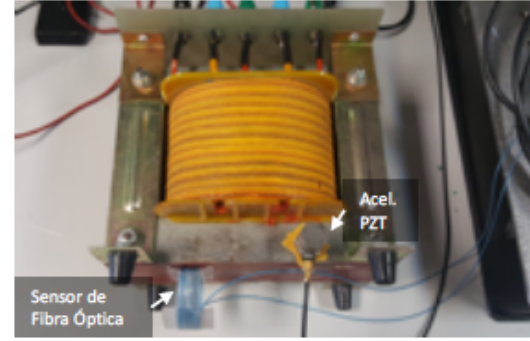
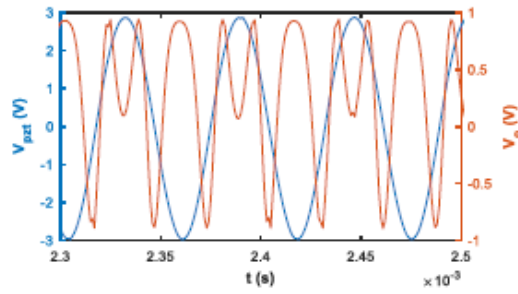


Fig. 6. Detalle fotográfico del transformador y los sensores de fibra óptica y acelerómetro piezoeléctrico para medida de vibraciones.

En la Fig. 7 se muestran los resultados obtenidos con el sensor de fibra óptica basado en un interferómetro con la detección sincrónica propuesta.

La Fig. 7a muestra la modulación en fase de $2\pi + \frac{\pi}{2}$ (equivalente a la modulación de $\pm \frac{\pi}{4}$) que permite adicionalmente obtener las cotas de máximos y mínimos de la función de transferencia del interferómetro; ver ecuación (2). Obsérvese que la escala de tiempos es un detalle para recoger 3,5 periodos de la portadora de 20 kHz.

La Fig. 7b recoge la fase óptica en cada instante (azul) calculada a partir de las dos señales muestreadas en los máximos y mínimos de la señal de modulación (V_{pzt}); por otra parte, también se presenta la fase total acumulada (naranja).

La Fig. 7c representa el desplazamiento (Δl) debido a las vibraciones mecánicas de acuerdo con (1), siendo aproximadamente de 100 nm como máximo.

Por último, en la Fig. 7d se puede observar el espectro de frecuencias de las vibraciones, observando un armónico fundamental a 100 Hz aproximadamente.

Por otra parte, en la Fig. 8 se muestran las vibraciones detectadas con el acelerómetro acondicionado para obtener una sensibilidad de 10 V/m/s². Obsérvese el ruido/interferencia de 50 Hz, y la notablemente menor sensibilidad comparada con el sensor de fibra óptica presentado anteriormente.

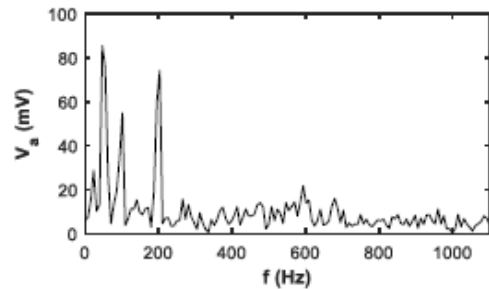


Fig. 8. Resultados de vibraciones con acelerómetro piezoeléctrico como base de comparación para analizar las prestaciones del sistema implementado de fibra óptica y detección sincrónica: espectro de vibraciones.

Fig. 7. Resultados experimentales de vibraciones del transformador con el sensor de fibra óptica en un interferómetro con detección sincrónica (a) detalle de la portadora generada y la salida de interferencia, (b) señal de fase óptica demodulada, (c) señal de vibraciones (elongación de la fibra óptica), (d) espectro de vibraciones.

VI. CONCLUSIONES

Se ha estudiado una nueva aproximación para demodular las señales de interferencia en sensores de fibra óptica de alta sensibilidad. Sus principales ventajas son las asociadas a una detección sincrónica (detección de fase) y la versatilidad de implementación electrónica (con circuitos de muestreo sincrónico, con microcontroladores o con sistemas de adquisición).

La detección sincrónica en cuadratura, junto con una salida de más de un ciclo que permite normalizar la amplitud, proporcionan una lectura de la fase óptica sin ambigüedad.

Se han presentado resultados experimentales con un sensor de vibración para transformadores en el rango de frecuencias típicas de 100 Hz y armónicos.

AGRADECIMIENTOS

El autor agradece al ministerio de Educación, Cultura y Deporte de España por la beca FPU 2016 con SIA: 998758.

REFERENCIAS

- [1] J.A. García-Souto, and H. Lamela, "Comparative analysis of optical-fibre interferometric sensors versus accelerometers: application to vibrations inside high-power transformers", *Journal of Optics A: Pure and Applied Optics* 4, pp. 318-326, 2002.
- [2] T. Suzuki, T. Okada, O. Sasaki, and T. Maruyama, "Real-time vibration measurement using a feedback type of laser diode interferometer with an optical fiber", *Opt. Eng.* 36, pp. 2496-2502, 1997.
- [3] J. A. Bucaro, and N. Lagakos, "Lightweight fiber optic microphones and accelerometers", *Rev. Sci. Instrum.* 72, pp. 2816-2821, 2001.
- [4] C. H. Chen, G. L. Ding, D. L. Zhang, Y. M. Cui, and S. Q. Li, "Michelson fiber optic accelerometer," *Rev. Sci. Instrum.* 69, pp. 3123-3126, 1998.
- [5] R.D. Pechstedt, and D.A. Jackson, "Performance analysis of a fiber optic accelerometer based on a compliant cylinder design", *Rev. Sci. Instrum.* 66, pp. 207-214, 1995.
- [6] A. Malki, P. Lecoy, J. Marty, C. Renouf, and P. Ferdinand, "Optical fiber accelerometer based on a silicon micromachined cantilever", *Appl. Opt.* 34, pp. 8014-8018, 1995.
- [7] J. M. López-Higuera, M. A. Morante, and A. Cobo, "Simple low-frequency optical fiber accelerometer with large rotating machine monitoring applications", *IEEE J. Lightw. Tech.* 15, pp. 1120-30, 1997.
- [8] J.A. García-Souto, H. Lamela-Rivera, "High resolution (<1nm) interferometric fiber-optic sensor of vibrations in high-power transformers", *Optics Express*, 14(21), pp. 9679-9686, 2006.
- [9] Wang Z, Hu Y, Meng Z, et al., "Working-point control method for readout of dynamic phase changes in interferometric fiber-optic sensors by tuning the laser frequency", *Appl Opt*, 47(19), pp. 3524-3529, 2008.
- [10] A.D. Kersey, D.A. Jackson and M. Corke, "Passive compensation Scheme suitable for use in single-mode fiber interferometers", *Electronics Letters*, vol. 16, pp. 392-393, 1982.
- [11] J.H. Cole, B.A. Danver and J.A. Bucaro, "Synthetic- heterodyne interferometric demodulation", *IEEE J. Quantum Electronics*, vol. 18, no. 4, pp. 694-697, 1982.
- [12] D.A. Jackson, A.D. Kersey, M. Corke and J.D.C. Jones, "Pseudo-Heterodyne detection scheme for optical interferometers", *Electronics Letters*, vol. 18, pp. 1192-1194, 1982.
- [13] M. J. Connelly, "Digital synthetic-heterodyne interferometric demodulation," *Journal of Optics A Pure & Applied Optics*, 2002, 4(6): S400-S405
- [14] A. Dandridge, A. B. Tveten, T. G. Giallorenzi, "Homodyne demodulation scheme for fiber optic sensors using phase generated carrier," *IEEE J. Quantum Electron.* QE-18, pp. 1647-1653, 1982.
- [15] J.A. García-Souto and H. Lamela, "Multi-channel Fiber-Optic Interferometric Sensor for Measurements of Temperature and Vibrations in Composite Materials", *IEEE Journal of Selected Topics on Quantum Electronics*, vol. 6, no. 5, pp. 780-787, 2000.